

UNIVERSIDAD DE SALAMANCA
FACULTAD DE CIENCIAS QUÍMICAS
DEPARTAMENTO DE QUÍMICA FÍSICA



Aplicación Computacional de un Nuevo Algoritmo Híbrido (ANN-AGDC) en Cinética Química



Sonsoles Encinar del Dedo

Salamanca, 2015

UNIVERSIDAD DE SALAMANCA
FACULTAD DE CIENCIAS QUÍMICAS
DEPARTAMENTO DE QUÍMICA FÍSICA



VNiVERSiDAD
D SALAMANCA

Aplicación Computacional de un Nuevo
Algoritmo Híbrido (ANN-AGDC)
en Cinética Química

Sonsoles Encinar del Dedo

Salamanca, 2015

Memoria presentada por la Licenciada Sonsoles
Encinar del Dedo para optar al Grado de Doctor

Salamanca, cinco de octubre de dos mil quince



VNiVERSiDAD
D SALAMANCA

JOSÉ LUIS GONZÁLEZ HERNÁNDEZ, PROFESOR TITULAR DE QUÍMICA FÍSICA Y M^a DEL MAR CANEDO ALONSO, PROFESOR CONTRATADO DOCTOR DE QUÍMICA FÍSICA DE LA UNIVERSIDAD DE SALAMANCA

CERTIFICAN que la investigación de doctorado “**Aplicación Computacional de un Nuevo Algoritmo Híbrido (ANN-AGDC) en Cinética Química**” se ha realizado bajo su dirección en este Departamento por la Licenciada en Química Dña. Sonsoles Encinar del Dedo.

En los términos establecidos por la normativa reguladora de los Estudios de Tercer Ciclo y Doctorado de la Universidad de Salamanca, se autoriza la presentación de este trabajo como investigación de Tesis Doctoral.

Salamanca, a 5 de Octubre de dos mil quince

Fdo. José Luis González Hernández

Fdo. M^a del Mar Canedo Alonso

Este trabajo se ha llevado a cabo en el Departamento de Química Física de la Universidad de Salamanca, y ha sido dirigido por los Drs. José Luis González Hernández (USAL) y M^a del Mar Canedo Alonso (USAL). Se agradece el uso de los medios computacionales del grupo.

AGRADECIMIENTOS

En primer lugar deseo expresar mi más sincero agradecimiento a los Drs. D. José Luis González Hernández y Dña. M^a del Mar Canedo Alonso, por elegirme, por confiar en mí y por darme la oportunidad de llegar a este momento. Su apoyo y orientación durante la realización del presente trabajo ha dado lugar a esta Tesis Doctoral que hubiese sido imposible sin ellos. Agradecer también el haberme facilitado siempre los medios suficientes para llevar a cabo todas las actividades propuestas durante el desarrollo de la Tesis.

A mis compañeros de laboratorio, que aunque permanentemente soy la única que ha estado ahí, el ir y venir de gente como Nieves, Almi, Raúl, y en especial Diana, (porque sin ella esto no habría sido ni la mitad de divertido y porque al fin y al cabo dos cabezas siempre piensan más que una) han hecho que esta Tesis tenga un poquito de todos.

A mis “amigas de Arévalo”, Leti, Tere, Esther, Isabelita, Laura...que aunque su pregunta más frecuente en estos 3 años ha sido: “¿Pero, y de qué sirve ser Doctora? A ver, entiéndeme, Doctora de las tuyas, no me refiero a los médicos, que ellos ya sé lo que hacen” en el fondo estoy convencida de que me quieren.

A mis “compañeras de piso”, Irene, Raquel, Mary, y en este último año también Belén, que han soportado mis agobios y siempre estaban dispuestas a sacarme de fiesta para despejar las confusiones y al día siguiente verlo todo de otra manera.

A mis “compañeros de carrera”, muchos de ellos todavía siguen por aquí y manifiestan su confianza en mí en todo momento, en especial Javi (conocido por todos como Apa), que da igual que esté en Salamanca, en Ourense, en Londres, en Béjar, de “Erasmus”, él siempre ha estado ahí para todo, da igual el día, la hora y el momento, además me ha ayudado en lo más complicado de la Tesis, la portada.

A Alfredo, que aunque este último año se ha hecho “madrileño” es otra de las personas que siempre me ha apoyado y me ha soportado en esta “montaña rusa” que ha sido la Tesis, por esas clases de Química Orgánica (impartidas junto con Víctor) en las cuales siempre encontraban una lógica a la reacción que yo fui incapaz de ver hasta que ellos me lo explicaban, y por muchísimas más cosas, simplemente GRACIAS.

Y, por último y más importante, a mi familia, en especial a mi madre. Sin ella nada de esto hubiese sido posible, por su ejemplo de superación ante las adversidades, su apoyo incondicional y su confianza plena en mí, que ha hecho que todo lo que soy y lo que llegue a ser en un futuro se lo deba a ella. TE QUIERO.

A mi madre

Índice

I. INTRODUCCIÓN.....	1
II. ANTECEDENTES BIBLIOGRÁFICOS	11
III. ASPECTOS TEÓRICOS.....	23
III.1. Algoritmos Híbridos	25
III.2. Redes Neuronales Artificiales	27
III.2.1. Sistema biológico	27
III.2.2. Redes Neuronales Artificiales.....	28
III.2.3. Características de las Redes Neuronales Artificiales	30
III.2.3.1. Topología de las Redes Neuronales	30
III.2.3.2. Conexión entre neuronas.....	32
III.2.3.3. Mecanismo de <i>Aprendizaje</i>	32
III.2.3.4. Tipo de Asociación entre las Informaciones de Entrada y Sa- lida.....	35
III.2.4. Tipos de Redes Neuronales Artificiales	36
III.2.4.1. <i>Perceptron</i>	36
III.2.4.2. <i>Adaline</i> y <i>Madaline</i>	37
III.2.4.3. Red <i>Backpropagation</i>	38
III.2.4.4. Red <i>Self Organizing Map</i>	41
III.2.4.5. Red <i>Counterpropagation</i>	41
III.2.4.6. Red <i>Hopfield</i>	42
III.2.4.7. Red <i>Bidirectional Associative Memory</i>	42
III.2.4.8. Red <i>Adaptive Resonance Theory</i>	42
III.3. Optimización Matemática. Aspectos Teóricos del AGDC	43
III.3.1. Aspectos generales de los métodos de gradiente.....	43
III.3.2. Algoritmo AGDC	49
III.3.3. Establecimiento y resolución del sistema de ecuaciones diferenciales de velocidad	50
III.3.4. Derivación Numérica	55
III.3.5. Cálculo de la matriz inversa	55

III.4. Aspectos Teóricos en Cinética Química	56
III.4.1. Aspectos Teóricos en Cinética Química Isotérmica	58
III.4.2. Aspectos Teóricos en Cinética Química No Isotérmica	65
III.4.2.1. Determinación de A y E. Ecuación de Arrhenius.....	66
III.4.2.1.1. Resolución exacta	67
III.4.2.1.2. Resolución numérica.....	68
III.4.2.2. Determinación de ΔH^\ddagger y ΔS^\ddagger . Ecuación de Eyring	70
 IV. ASPECTOS COMPUTACIONALES	 73
 IV.1. Diseño de ANN. Software informático	 75
IV.1.1. Ajuste de funciones en ANN	76
IV.1.1.1. Generación de datos	77
IV.1.1.1.a. Datos concentración/tiempo	77
IV.1.1.1.b. Datos absorbancia/tiempo	77
IV.1.1.1.c. Datos fracción remanente/tiempo.....	77
IV.1.1.2. Representaciones gráficas	78
IV.1.2. Diseño de la arquitectura de la ANN	79
IV.1.3. <i>Entrenamiento</i> o <i>Aprendizaje</i> de la Red Neuronal Artificial.....	79
IV.1.4. <i>Predicción</i>	80
IV.1.5. Procedimiento Computacional de MATLAB. Forma algorítmica	81
 IV.2. Aplicación computacional del algoritmo AGDC	 83
IV.2.1. Optimización de constantes de velocidad.....	86
IV.2.1.1. Establecimiento y resolución computacional del sistema de ecuaciones diferenciales de velocidad.....	86
IV.2.1.2. Cálculo computacional de las derivadas numéricas	89
IV.2.1.3. Inversión de la matriz Hessiana	92
IV.2.1.4. KINMODEL(AGDC)®Vs.1.0.....	94
IV.2.2. Optimización de Parámetros Termodinámicos de Activación mediante el algoritmo AGDC	97
IV.2.2.1. Cálculo computacional de las derivadas numéricas.....	98
IV.2.2.2. Inversión de la matriz Hessiana	100

IV.2.2.3. Algoritmo AGDC	100
IV.2.2.4. KINNOISOT(AGDC).....	102
V. DISEÑO DE EXPERIMENTOS Y ORGANIZACIÓN MATRICIAL DE LOS DATOS CINÉTICOS.....	107
V.1. Diseño de Experimentos (ED)	109
V.1.1. Diseño Experimental Factorial Completo (CFED).....	110
V.1.2. Diseño de Experimentos Central Compuesto (CCED).....	112
V.2. Organización matricial de curvas y datos cinéticos: Computación ANN y Representación gráfica (visualización).....	116
V.2.1. Tipos de datos cinéticos	117
V.2.2. Elección del tipo idóneo de datos cinéticos	118
V.2.3. Tipo de ordenación de las matrices de datos cinéticos isotérmicos y no isotérmicos.....	120
VI. INFLUENCIA DE DISTINTAS VARIABLES Y RESPUESTA DE LA RED NEURONAL.....	139
VI.1. Influencias de distintas variables en ANN	141
VI.1.1. Arquitectura de la Red Neuronal ANN. Capas de neuronas	142
VI.1.2. Capa de Entrada (<i>Input Layer</i>).....	142
VI.1.3. Capa/s Oculta/s (<i>Hidden Layer/s</i>).....	142
VI.1.4. Capa de Salida (<i>Output Layer</i>)	142
VI.2. Influencia en el Proceso de <i>Entrenamiento</i> o <i>Aprendizaje</i>	143
VI.2.1. Variables que intervienen en la Capa de Entrada (<i>Input Layer</i>)	143
VI.2.1.1. Número de datos de <i>Inputs</i> (n_d). Configuración de la capa de <i>Inputs</i>	143
VI.2.1.2. Número de curvas sintéticas (n_c) que se suministran a la Red ANN.....	143
VI.2.1.3. Variación en el porcentaje de reacción (conversión) en cada curva cinética	143

VI.2.1.4. Valores relativos de los porcentajes de distribución óptimos de <i>Training, Validation</i> y <i>Testing</i>	143
VI.2.1.5. Relaciones de valores $\varepsilon_1/\varepsilon_2/\varepsilon_3$ (Cinéticas Isotérmicas)	143
VI.2.2. Variables que intervienen en la Capa/s Oculta/s (<i>Hidden Layer</i>)	143
VI.2.2.1. Establecimiento del número óptimo de capas y el número óptimo de nodos o neuronas por capa/s	143
VI.2.3. Variables que intervienen en la Capa de Salida (<i>Output Layer</i>)	143
VI.2.3.1. Algoritmo de Optimización en el proceso de <i>Entrenamiento</i> ..	143
VII. RESULTADOS Y DISCUSIÓN	153
VII.1. Consideraciones Generales del Algoritmo Híbrido	155
VII.1.1. Redes Neuronales Artificiales	156
VII.1.1.1. Diseños de Experimentos.....	156
VII.1.1.2. <i>Entrenamiento</i> de la ANN. Valores Óptimos de la Arquitectura y de sus Variables.....	157
VII.1.1.3. <i>Predicción</i> de la Red Neuronal Óptima.....	159
VII.1.2. AGDC	161
VII.2. Determinación de Parámetros Termodinámicos de Activación en diferentes mecanismos de reacción mediante AH	163
VII.2.1. Análisis de las curvas T/t y de las curvas α/t	163
VII.2.1.1. Análisis de las curvas T/t	163
VII.2.1.2. Análisis de las curvas α/t	165
VII.2.2. Mecanismo I. $B_1 \xrightarrow{k_{12}} B_2$	167
VII.2.2.1. Determinación de los parámetros E_{12} y A_{12}	168
VII.2.2.1.1. Tratamiento de datos sintéticos.....	168
VII.2.2.1.1.a. <i>Predicción</i> de E_{12} y A_{12} mediante ANN.....	168
i) Diseño de Experimentos..	168
ii) <i>Entrenamiento</i>	169
iii) <i>Predicción</i>	172
VII.2.2.1.1.b. Optimización de E_{12} y A_{12} mediante AGDC	172

VII.2.2.1.2.	Tratamiento de datos experimentales	175
VII.2.2.1.2.a.	<i>Predicción</i> de E_{12} y A_{12} median- te ANN.....	176
	i) ED.....	176
	ii) <i>Entrenamiento</i>	177
	iii) <i>Predicción</i>	178
VII.2.2.1.2.b.	Optimización de E_{12} y A_{12} me- diante AGDC	179
VII.2.2.2.	Determinación de los parámetros ΔS_{12}^\ddagger y ΔH_{12}^\ddagger	180
VII.2.2.2.1.	Tratamiento de datos sintéticos.....	180
VII.2.2.2.1.a.	<i>Predicción</i> de ΔS_{12}^\ddagger y ΔH_{12}^\ddagger me- diante ANN.....	180
	i) Diseño de Experimentos..	180
	ii) <i>Entrenamiento</i>	181
	iii) <i>Predicción</i>	182
VII.2.2.2.1.b.	Optimización de ΔS_{12}^\ddagger y ΔH_{12}^\ddagger mediante AGDC	182
VII.2.2.2.2.	Tratamiento de datos experimentales	184
VII.2.2.2.2.a.	<i>Predicción</i> de ΔS_{12}^\ddagger y ΔH_{12}^\ddagger me- diante ANN.....	185
	i) ED.....	185
	ii) <i>Entrenamiento</i>	185
	iii) <i>Predicción</i>	187
VII.2.2.2.2.b.	Optimización de ΔS_{12}^\ddagger y ΔH_{12}^\ddagger mediante AGDC	187
VII.2.3.	Mecanismo II. $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$	190
VII.2.3.1.	Determinación de los parámetros E_{23} y A_{23}	190
VII.2.3.1.1.	Tratamiento de datos sintéticos.....	190
VII.2.3.1.1.a.	<i>Predicción</i> de E_{23} y A_{23} median- te ANN.....	190
	i) Diseño de Experimentos..	190
	ii) <i>Entrenamiento</i>	191
	iii) <i>Predicción</i>	193

VII.2.3.1.1.b. Optimización de E_{23} y A_{23} mediante AGDC	193
VII.2.3.1.2. Tratamiento de datos experimentales	195
VII.2.3.1.2.a. <i>Predicción</i> de E_{23} y A_{23} mediante ANN.....	196
i) Diseño de Experimentos..	196
ii) <i>Entrenamiento</i>	197
iii) <i>Predicción</i>	199
VII.2.3.1.2.b. Optimización de E_{12} y A_{12} mediante AGDC	199
VII.2.3.2. Determinación de los parámetros ΔS_{23}^\ddagger y ΔH_{23}^\ddagger	201
VII.2.3.2.1. Tratamiento de datos sintéticos.....	201
VII.2.3.2.1.a. <i>Predicción</i> de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger mediante ANN.....	201
i) Diseño de Experimentos..	201
ii) <i>Entrenamiento</i>	202
iii) <i>Predicción</i>	203
VII.2.3.2.1.b. Optimización de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger mediante AGDC	203
VII.2.3.2.2. Tratamiento de datos experimentales	206
VII.2.3.2.2.a. <i>Predicción</i> de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger mediante ANN.....	206
i) Diseño de Experimentos..	206
ii) <i>Entrenamiento</i>	206
iii) <i>Predicción</i>	208
VII.2.3.2.2.b. Optimización de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger mediante AGDC	208

VII.3. Determinación de constantes de velocidad mediante ANN. Mecanismo II.

$B_1 \rightarrow B_2 \rightarrow B_3$	211
VII.3.1. Tratamiento de datos sintéticos	211
i) Diseño de Experimentos	211
ii) <i>Entrenamiento</i>	212
iii) <i>Predicción</i>	215

VII.3.2. Tratamiento de datos experimentales	217
i) Diseño de Experimentos	220
ii) <i>Entrenamiento</i>	221
iii) <i>Predicción</i>	222
VII.4. Discriminación de mecanismos de reacción	223
VII.4.1. Discriminación de mecanismos de reacción mediante ANN	223
VII.4.1.1. Capacidad de Discriminación de mecanismos en modelos “Soft Modelling”	223
VII.4.1.2. Resultados de la discriminación mediante ANN.....	225
VII.4.1.2.1. Modelo I	225
VII.4.1.2.2. Modelo II	229
VII.4.1.2.3. Modelo III.....	232
VII.4.2. Discriminación de mecanismos de reacción mediante AGDC	235
VII.4.2.1. Determinación Constantes velocidad mediante la aplica- ción del algoritmo AGDC. Discriminación de mecanismos de reacción	235
VII.4.2.2. Resultados de la discriminación mediante AGDC	235
VIII. CONCLUSIONES	241
IX. REFERENCIAS BIBLIOGRÁFICAS	248
ANEXO	261

Acrónimos

AH	Algoritmo Híbrido
AG	Algoritmo Genético
AGDC	Algoritmo General de Descenso Controlado
AGDC(MW)	Algoritmo General de Descenso Controlado <i>Multiwavelength</i>
ANN	<i>Artificial Neural Networks</i>
ANN-MFCD	<i>Artificial Neural Networks-Modified Face Centered Design</i>
ART	<i>Adaptive Resonance Theory</i>
BAM	<i>Bidirectional Associative Memory</i>
CCED	<i>Central Composite Experimental Design</i>
CCR	<i>Classical curve resolution</i>
CFED	<i>Complete Factorial Experimental Design</i>
CQC	Cinética Química Computacional
CSCED	<i>Central Star Composite Experimental Design</i>
DE	Dominio Experimental
DMM	<i>Dynamic Multicomponent Mixtures</i>
DSC	<i>Differential Scanning Calorimetry</i>
DTA	<i>Differential Thermal Analysis</i>
DTG	<i>Derivative Thermogravimetry</i>
ED	<i>Experimental Design</i>
ED-ANN	<i>Experimental Design-Artificial Neural Networks</i>
EDCSC	Diseños de Experimentos Compuestos
EFA	<i>Evolving Factor Analysis</i>
FCD	<i>Face Centered Design</i>
FFED	<i>Factorial Fractional Experimental Design</i>
FORTTRAN	<i>FORmula TRANslating System</i>
GC	Gradientes Conjugados
GRAM	<i>Generalized Rank Annihilation Method</i>
GUI	<i>Graphical User Interface</i>
GUIDE	Editor de Interfaces de Usuario
HDNA-GA	<i>Hybrid Deoxyribonucleic Acid-Genetic Algorithm</i>
ICTAC	<i>International Confederation for Thermal Analysis and Calorimetry</i>
IUPAC	<i>International Union of Pure and Applied Chemistry</i>
LM	<i>Levenberg Marquardt</i>
LM-PAR	<i>Levenberg-Marquardt/PARAFAC</i>

LMS Error	<i>Least Mean Squared Error</i>
LU	<i>Lower-Upper</i>
MATLAB	<i>MATrix LABoratory</i>
MCBGA	<i>Namely Cultural-Based Genetic Algorithm</i>
MCR-ALS	<i>Multivariate Curve Resolution–Alternating Least Squares</i>
MFCD	<i>Modified Face Centered Design</i>
MLR	<i>Multiple Lineal Regression</i>
MSE	<i>Mean Squared Error</i>
MW	<i>Multiwavelenght</i>
ODE	<i>Ordinary Differential Equations</i>
PCA	<i>Principal Component Analysis</i>
PCR	<i>Principal Component Regresión</i>
PE	<i>Process Element</i>
PLS	<i>Partial Least Square</i>
PTA	<i>Parámetros Termodinámicos de Activación</i>
QN	<i>Quasi-Newton</i>
RMS	<i>Root Mean Squared</i>
RSM	<i>Response Surface Methodology</i>
SCED	<i>Star Composite Experimental Design</i>
SD	<i>Standard Deviation</i>
SDM	<i>Steepest Descent Method</i>
SED	<i>Star Experimental Design</i>
SIMULINK	<i>Plataforma de Simulación Multidominio</i>
SMM	<i>Static Multicomponent Mixtures</i>
SOM	<i>Self Organizing Map</i>
SQD	<i>Squared Quadratic Deviations</i>
SW	<i>Unilongitud de onda</i>
TCF	<i>Traditional Curves Fitting</i>
TG	<i>Thermogravimetry</i>
TR	<i>Training</i>
TS	<i>Testing</i>
VL	<i>Validation</i>
XRD	<i>X-ray Diffraction</i>

El guerrero de la luz no tiene miedo de llorar antiguas penas,
ni de alegrarse con nuevos descubrimientos.
Cuando siente que llegó el momento,
lo abandona todo y parte hacia su aventura tan soñada.
Manual del guerrero de la luz

I. Introducción

Capítulo I.

Introducción

La gran potencia, robustez y versatilidad que posee el software actual en el que se aplican los distintos métodos numéricos computacionales, hace posible el tratamiento y resolución de todo tipo de funciones matemáticas que correlacionan variables y parámetros de sistemas químicos físicos, pudiéndose acceder al conocimiento de los valores de los parámetros. Es más estos parámetros pueden determinarse incluso en aquellos casos en que no sea conocida o no exista una función matemática explícita de correlación entre variables y parámetros, siempre y cuando se disponga de amplios colectivos de datos experimentales. Existe una amplia variedad de software disponible para su uso y aplicación que extiende y facilita las posibilidades de aplicación incluso en el caso de usuarios no expertos.

Por su parte, la tecnología moderna permite en la actualidad la creación de grandes contingentes o bases de datos que requieren ser analizados para buscar una información refinada. Esto constituye un gran reto, pues requiere un desarrollo apropiado de metodologías robustas capaces de procesar estos grandes volúmenes de datos para convertirlos en conocimiento útil del sistema o del proceso en estudio, que permita una adecuada toma de decisiones predictivas. Se trata de realizar análisis prospectivos automatizados de los datos de los grandes dispositivos de almacenamiento (*data warehouse*) para la extracción de información oculta y predecible. De esta manera, han emergido nuevas disciplinas para abordar con garantías este problema, destacando entre otras las siguientes: *Data Mining* (“Minería de Datos”); *Knowledge Discovery* (“Descubrimiento del Conocimiento”), *Intelligent Data Analysis* (“Análisis Inteligente de Datos”); *Exploratory Data Analysis* (“Análisis Exploratorio de Datos”). Estas disciplinas utilizan técnicas del campo de la Inteligencia Artificial, fundamentalmente de Redes Neuronales Artificiales, (*Artificial Neural Networks*; ANN) que junto con los métodos de Optimización Matemáticos y Estadísticos, pueden implementarse en un software con un gran impacto en tareas como son la planificación económica, vigilancia e inteligencia empresarial, análisis financiero, análisis de mercados,...etc.

Como podemos apreciar, este análisis de datos persigue unos objetivos de toma de decisiones predictivas en ámbitos de tipo mercantil, empresarial, financiero, etc. en los que existe un número muy elevado de variables en las correlaciones, que es imposible plasmar en una expresión matemática. En nuestro caso, los objetivos aparentemente están alejados de éstos, pues consisten en la *Predicción* o determinación cuantitativa de parámetros de procesos que ocurren en Cinética Química. Sin embargo, tenemos en común que la evaluación y el análisis, lo efectuamos también sobre grandes colectivos de datos y las metodologías de tratamiento son idénticas, pues utilizamos técnicas del campo de la Inteligencia Artificial (Redes Neuronales

Artificiales; ANN) junto con un Algoritmo General de Descenso Controlado (AGDC), implementados en un software de similares características. Las diferencias son notables, puesto que: *a)* nuestra base de datos está compuesta por un enorme colectivo de datos exclusivamente numéricos, que nos va a permitir predecir valores estrictamente numéricos y cuantitativos de los parámetros cinéticos, *b)* la metodología computacional de aplicación del software de Redes Neuronales Artificiales junto con AGDC, la hemos tenido que orientar y adaptar a determinaciones estrictamente numéricas. Hemos utilizado la aplicación específica de ANN en MATLAB relativa a "*Fitting Functions*", donde intervienen robustos algoritmos de optimización matemática y minimización, aplicados a las matrices de *Outputs* y de *Targets*, y *c)* el análisis Estadístico se centra en el Análisis de Residuales del ajuste y a la determinación de errores. Como veremos posteriormente en el Capítulo dedicado a la revisión de la bibliografía, existen muy pocos trabajos en que los objetivos coincidan con los nuestros, es decir, en que se pretenda la *Predicción cuantitativa* de parámetros en general y cinéticos en particular, mediante un Algoritmo Híbrido (AH).

En el campo genérico de la Cinética Química se han producido grandes avances, debido sin duda a la reciente aplicación de metodologías y técnicas computacionales implementadas en el correspondiente software específico de alto nivel y que configura la Cinética Química Computacional (CQC). La determinación de Parámetros Cinéticos y Termodinámicos de Activación, se realiza en sistemas cinéticos de cierta complejidad, mediante tratamientos "*Hard- y Soft-modelling*", al computar enormes colectivos de datos cinéticos. El tratamiento se realiza mediante técnicas de Regresión no lineal univariable o multivariable. En el primer caso se aplican algoritmos de optimización matemática, con robustez creciente a medida que aumenta el número de parámetros cinéticos a optimizar. Nuestro grupo de investigación ha trabajado en esta línea de investigación diseñando y aplicando con éxito un robusto algoritmo, (denominado AGDC) en muy diversas versiones.^(65-74,115-131) Lo hemos aplicado con éxito para la determinación de Parámetros Cinéticos y Termodinámicos de Activación (en cinéticas isotérmicas y no isotérmicas) y para la discriminación y toma de decisiones entre la idoneidad de diversos mecanismos de reacción. Otra forma de tratamiento cinético corresponde a la computación de datos cinéticos monitorizados a MW adquiridos con dispositivos "*diode array*", aplicando técnicas de *Evolving Factor Analysis* (EFA) y computacionales de *Multivariate Curve Resolution–Alternating Least Squares* (MCR–ALS) con metodologías *Hard-modelling*, *Soft-modelling* e *Híbridas*, que han permitido la determinación de parámetros cinéticos exclusivamente.

La Memoria a la que pertenece el presente trabajo, la hemos estructurado considerando 8 Capítulos. Después de la Introducción general correspondiente al presente Capítulo I, realizamos una actualización bibliográfica de los trabajos publicados con anterioridad que de alguna manera están relacionados con el nuestro, resaltando aquellos en los que se aplican

técnicas de AH en Cinética y Termodinámica Química y concretamente los que utilizan los métodos de ANN y AGDC (Capítulo II).

En el Capítulo III, consideramos todos aquellos aspectos de índole teórica. Lo hemos subdividido en 4 partes, considerando en primer lugar, los fundamentos teóricos en los que descansan los Algoritmos Híbridos (AH), en segundo lugar las técnicas de Redes Neuronales Artificiales que utilizamos, en tercer lugar, los aspectos relacionados con la Optimización Matemática y más concretamente con el Algoritmo General de Descenso Controlado (AGDC) y por último en cuarto lugar, consideramos el ámbito científico al que pertenecen los sistemas en estudio que no es otro que el correspondiente a la Cinética Química. En relación a los aspectos teóricos de la técnica de ANN, realizamos una descripción general de la estructura, arquitectura y topología, funcionamiento y aplicaciones de las Redes Neuronales. Destacamos las importantes posibilidades de aplicación de ANN y los logros que se consiguen destacando su tremenda versatilidad, eficacia y facilidad de utilización por usuarios no expertos. Enfocamos nuestra atención a las características, resultados, logros,...etc. de las aplicaciones ANN, cuando los objetivos a alcanzar son de tipo cuantitativo y especialmente en la *Predicción* de parámetros de funciones en el campo de la Química Computacional.

En el Capítulo IV, consideramos todos aquellos aspectos de naturaleza computacional, desarrollando dos grandes apartados; en el primero de ellos consideramos los aspectos computacionales del amplio software que manejamos, principalmente el que se refiere a la aplicación MATLAB ⁽⁶⁴⁾ de las técnicas de ANN explicando su manejo, sus logros, las causas de posibles fallos y consejos o advertencias para que esto no se produzca. En el segundo de ellos, consignamos la aplicación computacional del algoritmo AGDC, tanto para constantes de velocidad como para Parámetros Termodinámicos de Activación (PTA).

El Capítulo V lo hemos dividido en 2 partes. La 1ª parte (Apartado V.1) se refiere a la organización y sistematización de las curvas que debemos considerar para alcanzar un *Dominio Experimental* óptimo, que nos garantice un proceso de *Entrenamiento* de la Red Neuronal verdaderamente eficaz. Es decir, es necesario analizar las posibilidades máximas de variación de cada uno de los parámetros cinéticos, para diseñar la distribución idónea de todas las curvas o experiencias entre los límites de los intervalos máximos, de acuerdo con la amplia casuística que nos ofrecen los distintos *Diseños de Experimentos* disponibles. Teniendo en cuenta que para la realización de este trabajo es necesario tratar con un enorme colectivo de curvas y una gran cantidad de datos numéricos por curva, cuyo tratamiento computacional resulta tremendamente engorroso, es conveniente seguir rigurosamente 2 importantes pautas como son, el diseño de una ordenación matricial rigurosa y la visualización mediante el trazado de convenientes representaciones gráficas. Ambos aspectos, configuran los contenidos del Apartado V.2

En el Capítulo VI exponemos el trabajo que hemos llevado a cabo relativo a la influencia que determinadas variables ejercen en los resultados obtenidos al aplicar la técnica

ANN en la determinación de parámetros cinéticos de una reacción química (Apartado VI.1). El apartado VI.2 lo hemos desglosado considerando distintos sub-apartados con las diferentes influencias, clasificadas por la acción sobre los procesos de *Entrenamiento*, considerando individualmente las respuestas de cada una de las capas que constituyen la Red Neuronal. Es decir, evaluamos las influencias del número de curvas, número de puntos por curva, porcentaje *Training*, *Validation* y *Testing*, relación de magnitudes de los coeficientes de absorción molar...etc. Con esta base de conocimiento podemos ensayar el tratamiento de la Red a otros sistemas cinéticos más complejos, así como realizar la *Predicción* de nuevos parámetros característicos de los sistemas químicos objeto de estudio.

En el Capítulo VII consignamos el estudio que hemos realizado relativo a los procesos de *Entrenamiento* y *Predicción* por parte del Algoritmo Híbrido que corresponde a la obtención definitiva de los parámetros cinéticos. Lo hemos estructurado considerando el tipo de datos sobre los que se realiza la computación y también teniendo en cuenta el tipo de parámetros a optimizar diferenciando entre constantes de velocidad y Parámetros Termodinámicos de Activación. Hemos seguido la secuencia clásica de tratamiento esto es, en primer lugar datos sintéticos con error al azar (*noise*) de acuerdo a la distribución de *Gauss*, impuesto por nosotros con una magnitud del orden del error experimental intrínseco. Este Capítulo lo hemos dividido de acuerdo al tipo de datos sobre los que se hace el tratamiento y los parámetros a determinar. Así, la primera parte (Apartado VII.1) corresponde a las consideraciones generales del Algoritmo Híbrido (AH), la segunda parte (Apartado VII.2), computamos la determinación de los Parámetros Termodinámicos de Activación en diferentes mecanismos de reacción mediante AH. El Algoritmo Híbrido (AH), está constituido por dos métodos con fundamentos matemáticos diferentes cuya aplicación secuencial se realiza en dos etapas. En la primera se aplica un método del tipo "*soft modelling*" de *Artificial Neural Networks* (ANN) que no necesita utilizar estimaciones iniciales de los parámetros y es el encargado de determinar unos valores de los parámetros en las inmediaciones del óptimo global. Estos valores constituyen las estimaciones iniciales de un robusto y eficaz algoritmo de gradiente AGDC capaz de alcanzar el óptimo global deseado que garantice el éxito de la optimización directa de los valores de PTA. Además, la aplicación del AH nos proporciona otra ventaja adicional al poder optimizar directamente en una sola etapa dichos parámetros sin necesidad de la determinación previa de las constantes cinéticas. Para poder realizar este tratamiento necesitamos obtener datos cinéticos procedentes de experiencias cinéticas no isotérmicas, imponiendo una variación controlada de la temperatura a lo largo de la cinética de la reacción. El colectivo de datos cinéticos que se consigue en el procedimiento clásico isotérmico, es muy pequeño pues su número está limitado por las exigencias experimentales al tratarse de series de unas pocas experiencias isotérmicas (normalmente no superior a 10-12), realizadas cada una de ellas a una temperatura distinta. En el caso de las no isotérmicas, la realización de una única cinética replicada es suficiente, ya que,

es factible la obtención de un enorme colectivo de datos cinéticos. Esto supone grandes ventajas si consideramos el enorme ahorro de tiempo en el trabajo de laboratorio que se requiere para realizar las experiencias cinéticas así como el considerable ahorro económico que supone el menor consumo de reactivos. En la tercera parte, consideramos la computación tanto de datos sintéticos (con y sin *noise*) como experimentales, en orden a determinar las constantes de velocidad por parte de la Red Neuronal. Los datos numéricos que hemos monitorizado son de 3 tipos: *a*) absorbancia total de la mezcla reaccionante a la que contribuyen (habitualmente) las 3 especies, dependiendo de la zona del espectro que hayamos elegido para la adquisición de datos realizada a una o a varias longitudes de onda, *b*) datos de absorbancia individual de cada especie, calculados indirectamente a partir de los valores numéricos de absorbancia total, cuando son conocidos los coeficientes de absorción molar de las 3 especies a la/s longitud/es de onda de trabajo y *c*) datos de concentración de cada especie calculados indirectamente a partir de los datos individuales de absorbancias. Los datos que más frecuentemente hemos utilizado son los de absorbancia total pues a lo largo del trabajo hemos comprobado que son los que tienen un mejor comportamiento. Sin embargo, en algunas circunstancias, como es el caso en que los valores de los coeficientes de absorción molar de las 3 especies son muy similares y donde la absorbancia total experimenta una muy pequeña variación con el tiempo al considerar los tiempos inicial y final, la ANN no discrimina satisfactoriamente entre las distintas curvas de perfiles con muy suave pendiente. En esos casos, hemos acudido a la computación de datos calculados de concentración y/o de absorbancias individuales obteniendo satisfactorios resultados. Por último, en la cuarta parte, evaluamos la capacidad de discriminación de mecanismos de reacción mediante la aplicación de AH (ANN y AGDC), obteniendo excelentes resultados en ambos casos. Los modelos tomados para llevar a cabo el proceso de discriminación han sido aquellos que poseen tres especies y tres constantes de velocidad.

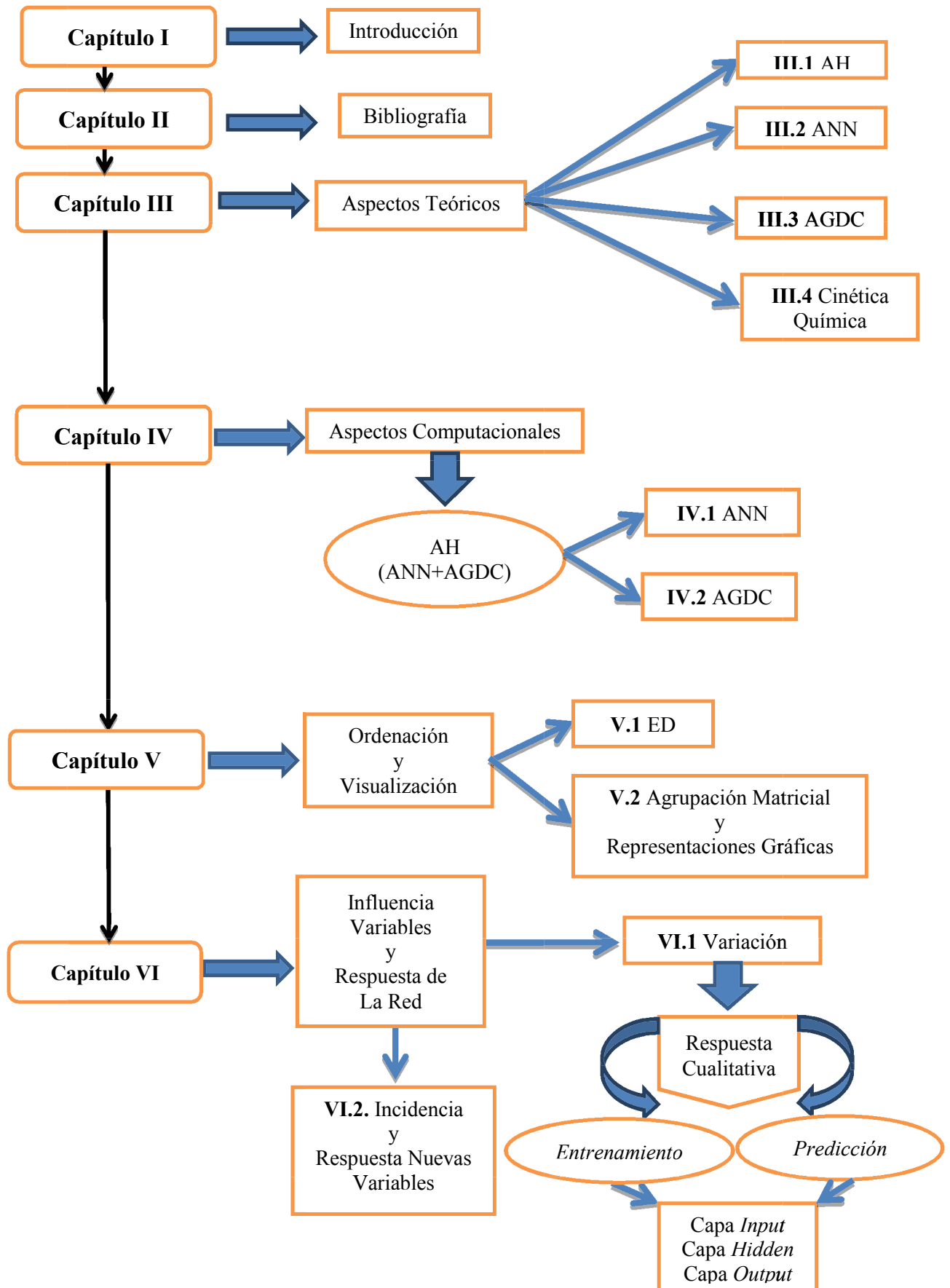
El Capítulo final (Capítulo VIII), recoge a modo de resumen las Conclusiones obtenidas a lo largo de la presente Memoria.

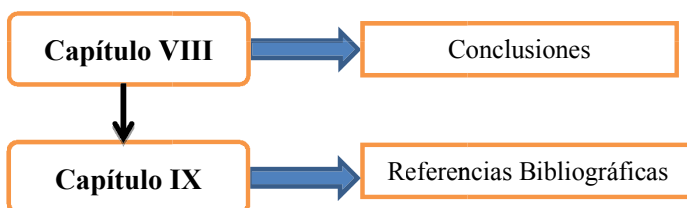
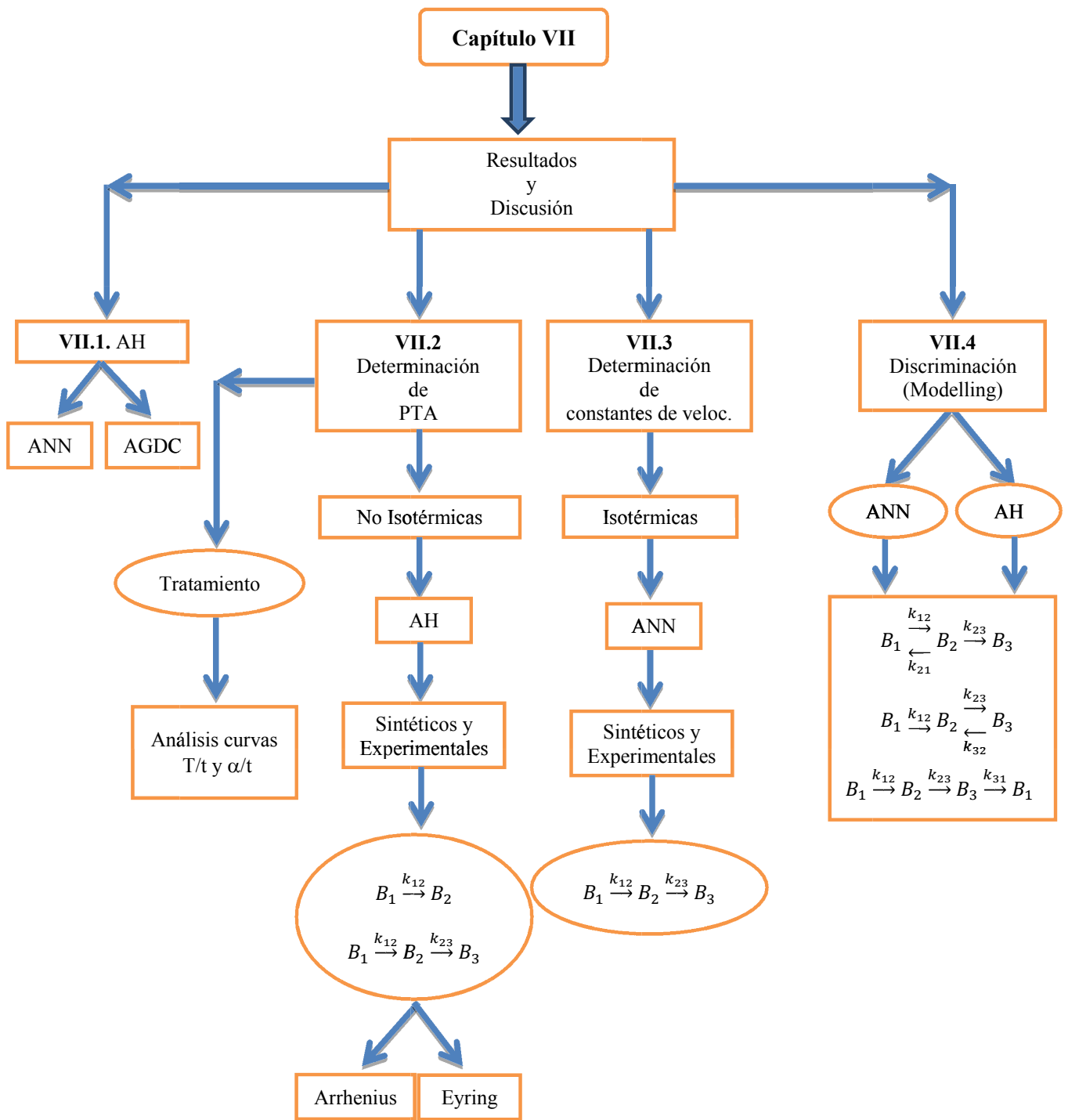
Por último, permítasenos hacer un comentario acerca de la redacción de la presente Memoria. Es un hecho constatable que no existe un pleno acuerdo entre los investigadores que desarrollan su actividad en los campos donde se aplica el método de Redes Neuronales Artificiales (ANN). Su gran versatilidad, permite la utilización por investigadores con muy diversas procedencias de muy distintos ámbitos de la Ciencia y de la Técnica. Se trata de un campo interdisciplinar en el que no existe una normativa externa que sirva para homogeneizar el argot utilizado en ANN, como por ejemplo lo hace IUPAC respecto a la Química. Se constata un doble problema: en general, hay varias posibilidades para la denominación de un mismo término incluso en lengua inglesa que es el idioma aceptado y comúnmente usado por todo el colectivo. A esto hay que añadirle que el conjunto de autores o investigadores castellano parlantes, unas veces conserva la denominación original en lenguaje inglés y en otras opta por

la traducción literal al castellano. Es evidente que esto puede causar problemas de comprensión para personas no especialistas o que no estén demasiado familiarizados con el argot utilizado. En el propio acrónimo ya puede aparecer la controversia. Así, la denominación Artificial Neural Networks (ANN) corresponde a la genuina denominación en lengua inglesa, mientras que si consideramos su traducción en castellano (Redes Neuronales Artificiales) le correspondería el acrónimo RNA (como algunos autores utilizan) que es la denominación universal por todos conocida del "Ribonucleic acid" (RNA). La traducción literal del inglés muchas veces podría llevar consigo una pérdida de la esencia inherente al término, como es el caso de "epochs" (iteraciones), "ill-conditioned matrix", "soft-modelling", "hard-modelling", "smoothing curves" ... y tantos otros que le harían perder precisión al sentido de los términos e incluso podrían resultar incoherentes o confusos.

La presente Memoria pertenece a un trabajo de Tesis Doctoral, con alta especificidad y que va dirigido principalmente a especialistas en la materia. Asimismo pretende tener una cierta vocación docente orientada al personal que realiza tareas de investigación en el campo ANN y que se encuentra en período de formación. Por tanto, hemos procurado redactarla siguiendo una pauta general que conserve el rigor que demanda el campo en el que se inscribe el trabajo, utilizando para los términos el nombre genuino y los acrónimos expresados en inglés y letra Cursiva y a continuación utilizamos la denominación en castellano. No obstante este orden puede alterarse muchas veces en las ocasiones que el sentido de la frase así lo aconseje.

Esquema I.1. Diagrama de flujo de la organización de la Memoria





*Los demás sólo aman y respetan al que se ama
y se respeta a sí mismo.
No intentes nunca agradar a todo el mundo,
o perderás el respeto de todos.
El manuscrito encontrado en Accra*

II. Antecedentes Bibliográficos

Capítulo II.

Antecedentes Bibliográficos

Un Algoritmo Híbrido (AH) es aquel que combina dos o más algoritmos distintos con el fin de resolver el mismo problema matemático. La bibliografía contiene referencias recientes en las que los autores han aplicado Algoritmos Híbridos (AH) en el campo de la Química y que esencialmente están constituidos por Algoritmos Genéticos (AG). El diseño y aplicación de este tipo de algoritmos permite resolver diversos problemas y revela el alto grado de fiabilidad y precisión en los resultados obtenidos. En esta línea R. Yao y col. ⁽¹⁾ desarrollan un Algoritmo Híbrido a partir de un Algoritmo Genético modificado mediante hibridaciones y estrategias de mutación. Este AH se utiliza para la obtención de parámetros cinéticos en un mecanismo de reacción complejo como es la síntesis de un determinado éter. De igual modo M. Maeder y col. ⁽²⁾ determinan las constantes de velocidad y de equilibrio en diversos mecanismos de reacción mediante la aplicación de un Algoritmo Híbrido basado en un Algoritmo Genético (AG). B. Saha y col. ⁽³⁾ utilizan Algoritmos Genéticos Híbridos para encontrar el mejor modelo y los parámetros cinéticos para la descomposición térmica de plásticos. De la misma manera Kan Dai y col. ⁽⁴⁾ emplean el Algoritmo Híbrido HDNA-GA para la estimación de parámetros en sistemas dinámicos, representados por modelos de reacción constituidos por procesos consecutivos o cíclicos. Recientemente A. Mostafa y col. ⁽⁵⁾ realizan el diseño de un reactor de tanque agitado y aplican el algoritmo MCBGA, basado en Algoritmos Genéticos, a varios procesos a optimizar. Resulta de gran interés la recopilación de diferentes aplicaciones en Quimiometría basadas en Algoritmos Genéticos realizada por A. Niazi y col. ⁽⁶⁾.

En bibliografía encontramos numerosas aplicaciones de AH en las que utilizan metodologías basadas en Redes Neuronales Artificiales junto con otros algoritmos como pueden ser los Algoritmos Genéticos. Así C. Hervás y col. ⁽⁷⁾ utilizan un método híbrido que consta de AG y Redes Neuronales Computacionales para seleccionar el número de entradas (*Inputs*) requeridas para corregir variaciones de temperatura apoyados en determinaciones cinéticas. Asimismo, A. C. Papes Filho y col. ⁽⁸⁾ aplican los AG junto con ANN para el diseño de un reactor a partir del cual llevaran a cabo la estimación de la velocidad de reacción en la oxidación del metanol a formaldehído industrial sobre catalizador de plata. De igual modo M. Huang y col. ⁽⁹⁾ desarrollan un Algoritmo Híbrido mediante combinación de Redes Neuronales Artificiales y Algoritmos Genéticos para simular el proceso de biodegradación de di-n-butil ftalato en un sistema anaeróbico / anóxico / óxico. Posteriormente G. Lisa y col. ⁽¹⁰⁾ utilizan Redes Neuronales y AG para predecir la termoestabilidad de derivados del ferroceno (compuesto organometálico con estructura tipo *sándwich* de Fe). Continuando en la misma línea M. Schubert y col. ⁽¹¹⁾ dedican su estudio a la utilización de Redes Neuronales y Algoritmos

Genéticos para mejorar la eficacia de un determinado proceso bioquímico enzimático. Es interesante destacar una serie de trabajos ⁽¹²⁻¹⁵⁾ que utilizan Algoritmos Híbridos que constan de Redes Neuronales Artificiales y otros Algoritmos como pueden ser los Genéticos. Estos Algoritmos Híbridos han sido utilizados para el estudio de procesos de adsorción, con el fin de estudiar procesos de eliminación de colorantes y sustancias peligrosas, eliminación de metileno por adsorción con carbón activo en aguas residuales, eliminación de tintes peligrosos en soluciones acuosas, y en la adsorción de tinte usando nanovarillas de óxido de zinc sobre carbón activo.

La metodología computacional basada en Redes Neuronales Artificiales (*Artificial Neural Networks*, ANN) es una poderosa herramienta que consta de un sistema de procesamiento de datos constituida por un largo número de elementos de sistemas simples, fuertemente interconectados y sistematizados, que tiene una arquitectura y funcionamiento basado en la estructura del cerebro. La metodología ANN es un procedimiento para el tratamiento de datos multivariable y multirrespuesta que se aplica en la actualidad en numerosos campos de la Ciencia, y en particular en el de la Química.

Resulta interesante destacar una serie de monografías ⁽¹⁶⁻¹⁹⁾ que recogen aspectos generales sobre la metodología ANN y que pueden ser de gran ayuda a la hora de entender la teoría fundamental del método y alguna de sus aplicaciones. La primera se debe a A. Freeman y D.M. Skapura ⁽¹⁶⁾ y recopila diferentes algoritmos, aplicaciones y técnicas de programación de carácter general. En la segunda sus autores J. Zupan y J. Gasteiger ⁽¹⁷⁾, recogen aspectos teóricos generales sobre la metodología y aplicaciones en el campo de la Química. Otras aportaciones bibliográficas más específicas, dedicadas al tratamiento de datos con fines Cinéticos y Analíticos ⁽¹⁸⁻¹⁹⁾, incluyen capítulos dedicados a la aplicación de ANN en estos campos.

Existen numerosas referencias en bibliografía que describen aplicaciones de ANN en campos muy diversos de la Ciencia como Medicina, Biología, Ingeniería Química, Farmacia, Biotecnología etc... En la mayoría de los casos se utiliza la metodología ANN para obtener conclusiones de tipo cualitativo, siendo muy pocos los casos en los que se utiliza para obtener resultados de índole cuantitativo. Destacaremos en primer lugar una serie de trabajos que nos permitirán ver las aplicaciones de ANN en campos muy diversos y posteriormente indicaremos las que se encuentran más relacionadas con nuestro campo de aplicación, la Cinética Química Computacional.

Uno de los campos de la Ciencia donde es frecuente la aplicación de ANN es en Farmacia como puede verse en el trabajo realizado por Y. Ni y col. ⁽²⁰⁻²¹⁾ utilizan ANN junto con otros métodos usados en Quimiometría para la determinación de diferentes fármacos como acetaminofen y fenobarbital en preparaciones farmacéuticas ⁽²⁰⁾ o bien pefloxacina y norfloxacina tanto en fármacos como en muestras de plasma humano ⁽²¹⁾. H. Goicoechea y

col.⁽²²⁾ estudian la monitorización espectrofotométrica de una serie de preparados farmacéuticos mediante ANN y otros métodos como PLS.

La metodología ANN ha sido ampliamente utilizada en Química en áreas muy diversas como es el caso de la Química Analítica. Así se ha utilizado para estimar constantes de equilibrio⁽²³⁾ tanto de protonación de sustancias como para el tratamiento de datos de equilibrio metal-ligando procedentes de técnicas potenciométricas, espectrofotométricas, extracción, etc... La metodología ANN también ha sido empleada para el estudio de fluidos, para predecir propiedades básicas como temperatura, presión y volumen crítico, masa molecular, etc..., de sustancias puras y fracciones de petróleo⁽²⁴⁾. También se han utilizado Redes Neuronales para el estudio de materiales como por ejemplo en el trabajo realizado por E.M. Bezerra y col.⁽²⁵⁾, en el que se determina la energía de activación de compuestos reforzados con fibras de carbono a partir de datos obtenidos por análisis Termogravimétrico y ANN. Otros trabajos utilizan ANN para el estudio de una relación estructura/actividad para reacciones en fase acuosa de compuestos orgánicos con radicales OH⁽²⁶⁾ o bien para la estimación de las constantes de velocidad para la reacción de hidrofluorocarbonos e hidrofluoroéteres con radicales OH⁽²⁷⁾.

Uno de los campos en los que ANN ha sido más utilizada es en el Análisis Químico, y más concretamente en la resolución de mezclas multicomponentes, como es el caso de los trabajos realizados por S. Ventura y col.⁽²⁸⁻³⁰⁾ en los que utilizan la metodología ANN para el estudio de diferentes mezclas multicomponentes binarias a partir de datos cinéticos. Dentro de esta línea M. Blanco y col.⁽³¹⁾ utilizaron ANN para la resolución de mezclas de enantiómeros a partir de datos cinéticos. Otros autores como G. López-Cueto y col.⁽³²⁾ analizan diferentes mezclas multicomponentes utilizando métodos de calibración multivariable lineales y no lineales.

Y. Ni y col.⁽³³⁻³⁸⁾ aplican la metodología ANN en la industria alimenticia, para la determinación cuantitativa simultánea de fructosa, glucosa y lactosa⁽³³⁾ en muestras de alimentos. Además estos mismos autores desarrollaron un método basado en ANN⁽³⁴⁾ para la determinación cualitativa simultánea de plaguicidas, Carbofuran, Isoprocarb y Propoxur en muestras de frutas y verduras. También han empleado la metodología ANN junto con otras técnicas para la determinación de hierro y aluminio en alimentos⁽³⁵⁾. En otros trabajos⁽³⁶⁾ utilizan ANN en combinación con otros métodos quimiométricos, para la determinación simultánea de yodato y peryodato a partir de datos cinéticos espectrofotométricos. Asimismo y mediante el empleo de diferentes técnicas quimiométricas y ANN, llevaron a cabo la determinación simultánea de maltol y etil maltol en alimentos⁽³⁷⁾. Además desarrollan un método para la determinación simultánea de 5 colorantes alimentarios artificiales⁽³⁸⁾, utilizando datos espectrofotométricos y métodos quimiométricos junto con ANN.

La aplicación de ANN para el análisis de mezclas multicomponentes ha sido llevada a cabo por otros autores como A. Safavi y col.⁽³⁹⁻⁴¹⁾, que analizan mezclas binarias formadas por

V(IV) y Fe(II) y Co(II) y V(IV), además de la determinación simultánea de carbidopa, levodopa y metildopa en presencia de citrato con la ayuda de técnicas de calibración multivariable y ANN. Otros autores ⁽⁴²⁾ realizan la determinación cinética simultánea de Cu(II), Fe(III) y Ni(II) mediante la transformada Wavelet y ANN. El análisis de mezclas multicomponentes mediante ANN se ha realizado a partir de datos procedentes de técnicas experimentales muy diversas como los métodos FIA, mediante este procedimiento se han determinado mezclas binarias de Cu(II) y Ni(II) ⁽⁴³⁾. Además se han podido determinar micro cantidades de Ir(III) ⁽⁴⁴⁾ en menas refinadas y aleaciones secundarias.

Las metodologías basadas en Redes Neuronales Artificiales pueden ser de gran ayuda en la investigación cinética de mecanismos de reacciones complejas, permitiendo determinar las etapas del proceso cinético (*modelización*). Una reacción compleja para cuyo estudio ha sido utilizado ANN ⁽⁴⁵⁾ es la oxidación de 2-octanol con ácido nítrico. Para el tratamiento de los datos experimentales procedentes de esta reacción se desarrolla un método basado en ANN, encontrándose que es un método eficaz para resolver problemas de *modelización*. La metodología ANN también ha sido utilizada para la *modelización* de reacciones catalíticas ⁽⁴⁶⁾ por lo que su aplicación se investiga para determinar diversos parámetros en diferentes procesos catalíticos y bajo distintas condiciones de reacción. De esta manera se aplica ANN para el estudio de reacciones catalíticas como la reducción de óxidos de nitrógeno, NO_x, con octano en exceso de oxígeno bajo Ag/Al₂O₃ ⁽⁴⁷⁾. La reacción de oxidación de CO ⁽⁴⁸⁾ catalizada por diferentes agentes también ha sido estudiada mediante la aplicación de ANN, demostrándose que la aplicación de esta metodología para la *modelización y predicción* en el campo de catálisis experimental es una nueva herramienta poderosa que puede acelerar el desarrollo y la optimización de la catálisis. De igual manera P. Valeh-e-Sheyda y col. ⁽⁴⁹⁾ utilizaron técnicas de Redes Neuronales para estimar la velocidad de reacción de la deshidratación del metanol en la síntesis catalítica de dimetil éter.

Los excelentes resultados que proporciona los métodos basados en Redes Neuronales Artificiales o bien combinándolo con otras técnicas en la *modelización* de sistemas químicos ha permitido su utilización para el tratamiento de sistemas enzimáticos ⁽⁵⁰⁻⁵⁵⁾, fundamentalmente en determinaciones cualitativas. Así por ejemplo ANN ha sido utilizada para la clasificación y *predicción* de la actividad enzimática de etilbenceno deshidrogenasa ⁽⁵⁰⁾ que cataliza la oxidación estereo-específica de etilbenceno y sus alcoholes derivados. En este trabajo se demuestra la aplicabilidad de este método para la *predicción* de la actividad de enzimas bioquímicas. La combinación de ANN con otros métodos quimiométricos como PCR (*Principal Component Regresión*), PLS (*Partial Least Squares*), permiten desarrollar un método de calibrado para la determinación de pesticidas a partir de métodos enzimáticos ⁽⁵¹⁾. D. Bas y col. ⁽⁵²⁻⁵⁴⁾ investigan la aplicabilidad de ANN en procesos bioquímicos. En un primer trabajo compararon las capacidades de cálculo de la metodología de superficie de respuesta (RSM) con

las Redes Neuronales Artificiales en una reacción enzimática (hidrólisis de maltosa) catalizada por amiloglucosidasa ⁽⁵²⁾. Investigaron la velocidad inicial de la reacción en función de la concentración del sustrato y el pH, utilizando dos diseños de experimentos diferentes, FCD y MFCD y dos técnicas de estimación diferentes, RSM y ANN. Mediante estos métodos determinaron las constantes cinéticas de la reacción enzimática, velocidad máxima de reacción (V_{max}) y la constante de Michaelis-Menten (K_m), a cinco pH diferentes, encontrando que los valores de las constantes cinéticas más cercanas a los verdaderos, se obtienen cuando se usan los datos de ANN-MFCD. Estos mismos autores investigaron la reacción de hidrólisis enzimática de la maltosa, catalizada por el enzima amiloglucosidasa ⁽⁵³⁾, utilizando ANN para la estimación de la velocidad de reacción enzimática, analizando el efecto en la velocidad de la reacción de la concentración de sustrato (maltosa) y producto (glucosa). Para esta misma reacción aplicaron la metodología basada en Redes Neuronales para estimar las constantes cinéticas ⁽⁵⁴⁾ comparando los resultados con los obtenidos por otros métodos convencionales. Otro ejemplo de la aplicabilidad de ANN en procesos bioquímicos es el trabajo realizado por J.A. Silva y col. ⁽⁵⁵⁾, en el que se desarrolla el modelo matemático para la síntesis enzimática de cefalosporina mediante la reacción de éster metílico de p-hidroxifenilglicina y ácido 6-aminopenicilánico catalizada por penicilina G acilasa inmovilizada en glutaraldehído-quitosano a 25°C y pH 6.5. Emplean modelos que acoplan ANN y ecuaciones de balance de masas, para reproducir el rendimiento del lote de reactores para la producción de amoxicilina. Este enfoque proporciona resultados precisos, sin estudiar el intervalo de concentración del sustrato.

La determinación simultánea del mecanismo de reacción y la estimación de las constantes de velocidad, es un aspecto muy importante en Cinética Química y en el que también se ha utilizado ANN ⁽⁵⁶⁾. Un problema importante en cinética química homogénea es la evaluación del mecanismo de reacciones químicas complejas. El proceso consiste en investigar todas las posibles especies que intervienen en la reacción, enumerar los pasos elementales y determinar la constante de velocidad de reacción de todos los pasos elementales. En este proceso pueden surgir numerosos problemas, generalmente en la medida de las concentraciones (pocos datos, no todas las especies son medibles,...), pudiendo existir una dependencia entre los parámetros con lo que sólo algunos de ellos pueden ser estimados. Intentando solventar estos problemas B. Kovacs y col. ⁽⁵⁶⁾ desarrollan dos métodos diferentes (uno de los cuales está basado en ANN) y los aplican al estudio de un mecanismo de reacción triangular cíclica formado por tres especies para la determinación de una serie de parámetros cinéticos. Otros autores como por ejemplo N.M. Halberstam y col. ⁽⁵⁷⁾ utilizan ANN para el estudio de la reacción de hidrólisis ácida de ésteres de ácidos carboxílicos con objeto de predecir relaciones entre la estructura de los compuestos, condiciones de reacción y constantes de velocidad.

La reacción de hidrólisis de 2,7-dicianonaftaleno responde a un mecanismo de dos reacciones consecutivas irreversibles, N. Lemes ⁽⁵⁸⁾ utilizan un método basado en ANN para la

determinación de constantes de velocidad a partir de datos cinéticos correspondientes al producto de la reacción y coeficientes de absorción molar a partir de datos de absorbancia.

La metodología ANN es un versátil método "*soft modelling*" que se aplica en muy diversos campos con buenos resultados⁽⁵⁹⁾. Sin embargo los resultados de la aplicación de ANN con fines cuantitativos en Cinética Química en modelos sencillos no siempre son aceptables. Ahora bien, es necesario realizar numerosos *Entrenamientos* de la Red Neuronal para la obtención de sus arquitecturas óptimas que hace que el método sea lento y tedioso y no se pueda utilizar individualmente en todos los casos. Este aspecto se destaca en el trabajo publicado por F. Amato y col.⁽⁶⁰⁾ en el que se obtienen los primeros resultados al realizar el tratamiento computacional con la aplicaciones de software informático, Trajan⁽⁶¹⁾ como simulador de Redes Neuronales Artificiales y STATISTICA⁽⁶²⁾, en las tareas de análisis Estadístico y de errores. Ambas aplicaciones se integran en una nueva metodología computacional ED-ANN que se aplica con éxito a datos cinéticos de concentración, obteniendo brillantes resultados. En esta misma línea J.L. González Hernández y col.⁽⁶³⁾, llevan a cabo la determinación de constantes de velocidad, desarrollando un procedimiento en el que en primer lugar se realiza un Diseño de Experimentos del tipo *Central Star Composite Experimental Design*, (CSCED) y posteriormente se lleva a cabo la *predicción* de las constantes de velocidad mediante ANN. En este trabajo se utilizó el software proporcionado por **MATLAB**⁽⁶⁴⁾ para el diseño de la Arquitectura de la Red, así como para realizar el proceso de *Entrenamiento y Predicción*. El procedimiento se aplica para la determinación de las constantes de velocidad de la reacción que tiene lugar entre el 2-mercaptoetanol con Carbonil Cianida 3-Clorofenilhidrazona ((3-Cl)-PHPD).

Uno de los objetivos del trabajo que recoge la presente Memoria es el desarrollo de un Algoritmo Híbrido para la determinación de Parámetros Termodinámicos de Activación (PTA) y constantes de velocidad de diferentes mecanismos de reacción. Este algoritmo está constituido por una combinación de dos algoritmos: el método "*soft modelling*" de Redes Neuronales Artificiales (ANN) y el Algoritmo General de Descenso Controlado (AGDC).

El algoritmo AGDC ha sido diseñado y aplicado con éxito en nuestro laboratorio en el tratamiento de muchos sistemas en el campo de la Química Física⁽⁶⁵⁻⁷⁵⁾. Se trata de un algoritmo de gradiente de segundo orden en el que se realiza un control continuo y exhaustivo de la dirección y sentido del vector de movimiento, evaluando cada etapa realizada en el procedimiento iterativo. El propio método consta de los dispositivos correctores que en caso necesario, pueden corregir y modificar automáticamente el vector de movimiento para asegurar el descenso al óptimo global. Además AGDC es muy resistente a la presencia de mínimos locales y superficies cuyas respuestas están mal condicionadas e incluye diversos métodos de representación gráfica (3D, mapas de contorno, etc.) que permiten tanto el seguimiento gráfico

de la evolución del proceso de optimización como la exploración en detalle de la región próxima al mínimo global para detectar la posible existencia de puntos singulares.

El precursor del algoritmo AGDC es el algoritmo **AGDC87** que fue inicialmente diseñado para la determinación exclusiva de las constantes termodinámicas macroscópicas de disociación a diferentes temperaturas de un conjunto de sustancias mono y bifuncionales de interés farmacológico ⁽⁶⁵⁾. Posteriormente se desarrolla el algoritmo **AGDC** basado en un tratamiento estrictamente numérico (cálculo de derivadas, vector gradiente, Hessiana, etc...) y se diseñaron una serie de aplicaciones computacionales para la determinación de diferentes parámetros. La primera de estas aplicaciones es el programa **KINAGDC** que fue aplicado para el tratamiento de la reacción de aminolisis de diferentes compuestos β -lactámicos ⁽⁶⁶⁾. Esta metodología fue modificada posteriormente para el tratamiento de datos cinéticos experimentales espectrofotométricos obtenidos a multilongitud de onda, desarrollando y aplicando el programa **KINAGDC-MW** para el tratamiento de los datos experimentales procedentes de la reacción de aminolisis en medio básico de carbenicilina y propilamina ⁽⁶⁷⁾.

El objetivo perseguido desde el inicio de la aplicación del algoritmo **AGDC** en cinética química, ha sido diseñar un programa generalizado que se adaptara a diferentes mecanismos de reacción capaz de determinar parámetros cinéticos y analíticos, sin necesidad de realizar ninguna modificación previa. Por este motivo y en vista de los resultados obtenidos hasta ese momento, se desarrolló el tratamiento computacional **ANALKIN(AGDC)@Vs.1.0** ⁽⁶⁸⁻⁶⁹⁾. Este programa propone un método generalizado que permite el tratamiento automático de cualquier mecanismo de reacción, ofreciendo la posibilidad de determinación de los siguientes parámetros:

- a) Constantes individuales de velocidad para cualquier mecanismo de reacción, conocida la composición inicial de la mezcla reaccionante.
- b) Concentraciones iniciales de las especies que intervienen en la reacción, conocidas las constantes de velocidad de las reacciones que tienen lugar.
- c) Determinación de forma simultánea y conjunta de las concentraciones iniciales de las especies que intervienen en la reacción y las constantes de velocidad de las reacciones que tienen lugar.

La versatilidad del programa se comprobó tras ser aplicado al análisis cuantitativo de mezclas multicomponentes de dos tipos:

- Mezclas Estáticas (SMM): aquellas cuya composición cuantitativa objeto del análisis, es invariable y constante. En ellas, conocida previamente la composición cualitativa, se adicionan varios reactivos que reaccionan con los analitos, evaluando los procesos cinéticos que tienen lugar lo que posibilita el conocimiento de la composición cuantitativa de la mezcla.

- Mezclas Dinámicas (DMM): aquellas cuya composición cuantitativa objeto del análisis, es variable con el tiempo, los analitos componentes de la mezcla, desempeñan todos ellos el papel de reactivos y/o productos de una reacción química que ha sido inducida por la adición de uno o más reactivos y cuya transformación cinético-química es evaluada computacionalmente en orden a determinar la composición cuantitativa de la mezcla inicial.

En un trabajo posterior J.L. González Hernández y col.⁽⁷⁰⁾ utilizaron el programa **ANALKIN(AGDC)@Vs.1.0** para el tratamiento de los datos experimentales procedentes de la reacción de isomerización de colest-5-en-3-ona a colest-4-en-3-ona catalizada por etóxido sódico. El procedimiento computacional empleado para esta reacción, se compara con el tratamiento clásico basado en la utilización del método aproximado del Estado Estacionario. Se determinan las constantes de velocidad, Parámetros Termodinámicos de Activación, constante de equilibrio y la Entalpía de la reacción de isomerización. El tratamiento computacional permite la determinación de la concentración de las especies que intervienen en el mecanismo, incluyendo 3,5-dienolato, como especie intermedia muy reactiva. También se discrimina entre posibles mecanismos para la reacción, validando aquel que comprende dos pasos reversibles.

El programa **ANALKIN(AGDC)@Vs.1.0** también ha sido utilizado posteriormente por J.L. González Hernández y col.⁽⁷¹⁾ para el tratamiento cinético y estudio del mecanismo de la reacción de oxidación del esteroide colest-4-en-3-ona mediante oxígeno molecular y catalizada por etóxido sódico. La aplicación del tratamiento computacional a esta reacción permite el cálculo de las concentraciones de todas las especies involucradas determinando cuál de los posibles mecanismos representa mejor los datos experimentales.

Se han desarrollado nuevas aplicaciones computacionales generalizadas del algoritmo **AGDC** como, por ejemplo, **KINMODEL(AGDC)**⁽⁷²⁻⁷³⁾. Se trata de un programa computacional *multipurpose* para el tratamiento de datos cinéticos, que permite la determinación de diferentes parámetros y es válido para el tratamiento automático de cualquier mecanismo de reacción. Mediante esta aplicación es posible la determinación, de forma individual o conjunta, de diferentes parámetros, como son constantes de velocidad, concentraciones iniciales y coeficientes de absorción molar. En estos trabajos se ha estudiado la optimización conjunta de parámetros de muy diferente naturaleza, analizándose en detalle la optimización de los coeficientes de absorción molar de las distintas especies de manera simultánea junto con las constantes individuales de velocidad⁽⁷²⁾. Asimismo se estudió la optimización conjunta de constantes de velocidad y concentraciones iniciales⁽⁷³⁾. En ambos casos se trata en profundidad el problema que se plantea en el proceso de optimización conjunta de parámetros de diferente naturaleza, en el caso de que exista más de un grupo de parámetros que nos proporcionen el mismo valor de la propiedad observada experimentalmente. Así, en el caso de la optimización conjunta de constantes de velocidad y coeficientes de absorción molar a

partir de datos de absorbancia total ⁽⁷²⁾, se produce un problema de ambigüedad en la solución, encontrándose dos soluciones posibles indiscernibles desde el punto de vista matemático. La optimización conjunta de constantes de velocidad y la concentración inicial de la especie intermedia conduce a un problema de ambigüedad si la optimización se lleva a cabo a partir de datos cinéticos correspondientes a la especie intermedia ⁽⁷³⁾. Posteriormente se aplican diferentes estrategias de optimización para la determinación de constantes de velocidad y coeficientes de absorción molar en el sistema de reacciones consecutivas irreversibles ⁽⁷⁴⁾. El procedimiento computacional combina 3 metodologías diferentes de optimización matemática basadas en el algoritmo AGDC(MW).

Tal y como hemos señalado uno de los objetivos del trabajo que recoge la presente Memoria es la aplicación de un Algoritmo Híbrido AH (ANN-AGDC) para el tratamiento de cinéticas no isotérmicas y la determinación de Parámetros Termodinámicos de Activación. A continuación consideramos una serie de aportaciones bibliográficas sobre cinéticas no isotérmicas que presenta gran interés en Cinética Química por lo que se han estudiado distintos aspectos de su tratamiento.

M. E. Brown y col. ⁽⁷⁵⁾ estudian la determinación de los parámetros de Arrhenius, energía de activación, (E) y el factor preexponencial, (A) mediante experimentos realizados en diferentes condiciones. En primer lugar realizan experiencias cinéticas isotérmicas a diferentes temperaturas, determinando a partir de los resultados obtenidos E y A . Posteriormente estudian la posibilidad de realizar un único experimento no isotérmico, para la determinación de los parámetros de Arrhenius, analizando las diferentes posibilidades de abordar el problema y las ventajas e inconvenientes de cada uno de los métodos.

En el Departamento de Química Física de la Universidad de Salamanca, se ha estudiado en profundidad el tratamiento de Cinéticas no isotérmicas, analizando una serie de situaciones que pueden tener lugar en el estudio de las mismas ⁽⁷⁶⁻⁷⁸⁾. J. L. González Hernández y col. ⁽⁷⁶⁾ desarrollan un método para la determinación de los parámetros de la ecuación de Arrhenius, obteniendo excelentes resultados en la precisión del método. En otro trabajo posterior ⁽⁷⁷⁾ proponen un método para el tratamiento de los datos en cinéticas no isotérmicas, con variación no lineal de temperatura con respecto al tiempo y realizan un estudio comparativo entre este método y aquel en el que la variación es lineal para el caso de una determinada reacción química en disolución. Estos mismos autores ⁽⁷⁸⁾ proponen un experimento como Práctica docente para un laboratorio de pregrado, en el que se lleva a cabo un estudio Cinético no isotérmico, determinando los parámetros de activación mediante experimentos con variación hiperbólica de temperatura.

G. Alibrandi ⁽⁷⁹⁾ obtiene constantes cinéticas y parámetros de activación en algunas reacciones inorgánicas basándose en un método en el que los datos cinéticos son obtenidos espectrofotométricamente variando la temperatura. M. Maeder y col. ⁽⁸⁰⁾ analizan el tratamiento

de Cinéticas no isotérmicas y proponen un algoritmo computacional en el que la integración numérica se lleva a cabo mediante algoritmos como Runge-Kutta o Gear determinándose los parámetros de activación. Steven C. Hodgson y col. ⁽⁸¹⁾ estudian la reacción de oxidación del ácido fórmico con bromo para la determinación de la energía de activación en cinéticas químicas no isotérmicas. K.T. Wojciechowski y col. ⁽⁸²⁾ desarrollan un programa de simulaciones numérico para la modelización de procesos químicos en condiciones no isotérmicas determinando la energía de activación y el orden de la reacción utilizando los datos de TG y DTA (DSC) simultáneamente. Erik Furusjö y col. ⁽⁸³⁾ estiman parámetros cinéticos a partir de condiciones no isotérmicas monitorizando las reacciones orgánicas espectrofotométricamente. Ambos trabajos se inscriben en la técnica del Análisis Térmico Diferencial.

Otros autores como M. Maeder y col. ⁽⁸⁴⁾ analizan en un trabajo *tutorial* el tratamiento de Cinéticas no isotérmicas y proponen un algoritmo computacional en el que la integración numérica se lleva a cabo mediante algoritmos como Runge-Kutta o Gear determinándose los parámetros de activación. De igual manera en otro trabajo posterior ⁽⁸⁵⁾ utilizan MATLAB ⁽⁶⁴⁾ para la obtención de resultados basados en la *modelización* de procesos químicos, fundamentalmente en el cálculo de los perfiles de concentración de todas las especies en función del tiempo, incluyendo ejemplos de reacciones a temperatura y pH no constantes.

Posteriormente L.J. Rodríguez-Aragón y col. ⁽⁸⁶⁾ realizaron un diseño para la optimización de los parámetros de la ecuación de Arrhenius con una gran eficacia y precisión para cada parámetro en particular.

En bibliografía se pueden encontrar estudios cinéticos no isotérmicos de reacciones químicas concretas. Así, destacamos el trabajo realizado por J.Hong y col. ⁽⁸⁷⁾ en el que realizaron un estudio no isotérmico sobre la descomposición del acetato de níquel tetrahidratado en aire usando técnicas como (TG)-DTG, DSC y XRD, donde la descomposición ocurre en dos pasos y el producto final es NiO. En esta línea C. A. Ribeiro y col. ⁽⁸⁸⁾ llevan a cabo el estudio de la cinética no isotérmica de la oxidación del carburo de tungsteno.

Otros estudios ⁽⁸⁹⁻⁹⁰⁾ nos desvelan una serie de recomendaciones desarrolladas por el ICTAC para realizar cálculos computacionales cinéticos cuando se trabaja con datos de Análisis Térmico, en cinéticas no isotérmicas, etc.

Por último, permítasenos destacar dos publicaciones ⁽⁹¹⁻⁹²⁾ resultantes del trabajo que recoge la presente Memoria y que ha sido realizado en el Departamento de Química Física de la Universidad de Salamanca. En estos trabajos se determinan los Parámetros Termodinámicos de Activación a partir de datos procedentes de cinéticas no isotérmicas para varios modelos cinéticos, mediante la aplicación de un Algoritmo Híbrido. Este algoritmo consta de dos metodologías: Redes Neuronales Artificiales y el algoritmo de optimización matemática AGDC.

*Podemos perdonar fácilmente a un niño que teme la oscuridad,
la verdadera tragedia es cuando los hombres temen la luz.*
Platón

III. Aspectos Teóricos

Capítulo III.

Aspectos Teóricos

En el presente Capítulo se exponen de una manera ordenada los fundamentos teóricos en los que se fundamentan las metodologías que hemos utilizado en nuestro trabajo para la determinación de Parámetros Cinéticos y Termodinámicos de Activación:

- En un primer apartado (Apartado III.1) se explican los aspectos teóricos de los Algoritmos Híbridos (AH), que son aquellos que combinan diferentes metodologías para la resolución de un mismo problema. En nuestro caso el AH está formado por la combinación de la metodología que utiliza Redes Neuronales Artificiales (ANN) y el Algoritmo de Optimización Matemática AGDC.

- Puesto que una de las metodologías utilizadas en nuestro trabajo se basa en la aplicación de Redes Neuronales Artificiales (ANN), en el Apartado III.2 exponemos los fundamentos teóricos de las mismas.

- El Apartado III.3 está dedicado a los Algoritmos de Optimización Matemática, centrándonos en los métodos de gradiente y en concreto exponemos las bases teóricas del Algoritmo Generalizado de Descenso Controlado (AGDC).

- Por último, dedicamos otro apartado a la exposición de los aspectos teóricos de Cinética Química: Cinéticas isotérmicas y No isotérmicas (Apartado III. 4).

III.1.- Algoritmos Híbridos (AH)

Un Algoritmo Híbrido (AH) es aquel en el que dos o más algoritmos distintos se combinan con el fin de resolver el mismo problema matemático. La aplicación de un AH se puede realizar de diferentes maneras, teniendo en cuenta el tipo de datos. Una primera forma de aplicación del AH sería mediante la elección del algoritmo que se adapta mejor a los datos. Una segunda forma de aplicación del AH consistiría en el cambio de algoritmo durante el proceso de determinación de los parámetros, de acuerdo con las condiciones del proceso. Este procedimiento se lleva a cabo generalmente para combinar las características deseadas de cada uno de los algoritmos, de manera que el algoritmo general es mejor que los componentes individuales ⁽⁹³⁾. El Algoritmo Híbrido es capaz de incorporar diferentes técnicas dando lugar a un algoritmo que es una combinación de todos los procedimientos y que cosecha mejores resultados que los algoritmos individuales. Esta segunda forma se corresponde con el AH que se propone en este trabajo. El objetivo final es llegar a la combinación de las características más apropiadas de cada uno, por lo que el AH general global puede funcionar mejor que cada uno de los métodos individualmente.

El AH no sólo se refiere a una combinación de múltiples algoritmos para resolver un problema diferente. De hecho, muchos algoritmos pueden considerarse como combinaciones de algoritmos sencillos. El AH es una combinación de algoritmos que resuelven el mismo problema, pero difieren en sus características, en particular en la eficiencia en su aplicación.

A lo largo del trabajo que recoge la presente Memoria se ha aplicado un Algoritmo Híbrido para la determinación de Parámetros Termodinámicos de Activación (PTA) y constantes de velocidad de varios mecanismos. Este algoritmo está constituido por una combinación de dos algoritmos que se basan en diferentes principios matemáticos y que se aplican secuencialmente. En una primera etapa, se aplica el método "*soft modelling*" de Redes Neuronales Artificiales (ANN) para la determinación de los Parámetros Termodinámicos de Activación (PTA) y de las constantes de velocidad. En una segunda etapa, los valores de los parámetros obtenidos mediante la aplicación de ANN se utilizan como estimaciones iniciales del algoritmo de optimización (AGDC) aplicado en una segunda etapa con el objetivo de mejorar los valores finales de los parámetros.

Tal y como hemos señalado, el AH que se utiliza a lo largo de este trabajo, comprende dos algoritmos basados en diferentes principios matemáticos y cuya aplicación se lleva a cabo de manera secuencial. El método "*soft modelling*" que se aplica mediante el uso de Redes Neuronales Artificiales (ANN), presenta una gran ventaja sobre otros métodos de determinación de parámetros ya que no es necesario partir de estimaciones iniciales de los parámetros a determinar. Los métodos de optimización matemática necesitan valores estimados de los parámetros a determinar, además es conveniente que estas estimaciones sean cercanas a los valores reales de los parámetros para garantizar el éxito del proceso de optimización. El AH desarrollado a lo largo de este trabajo determina en primer lugar unos valores de los parámetros mediante la aplicación de ANN y estos valores, se utilizan como estimaciones iniciales de un algoritmo de optimización matemática AGDC. Este algoritmo es un método numérico de optimización matemática de gradiente de segundo orden, eficiente y robusto, capaz de alcanzar el mínimo global deseado que garantiza el éxito de la optimización final de los parámetros.

El AH que utiliza la metodología ANN y AGDC ofrece una serie de ventajas importantes sobre los métodos tradicionales:

- No es necesario conocer *a priori* las estimaciones iniciales de los parámetros a determinar. En una primera etapa se aplica la metodología ANN para la determinación de los parámetros, que como hemos señalado, este método no requiere estimaciones iniciales de los mismos. Los parámetros determinados mediante la metodología ANN se utilizan como estimaciones iniciales en la aplicación del algoritmo AGDC.
- La mayor robustez del AH disminuye la probabilidad de alcanzar puntos singulares tales como mínimos locales, puntos de silla, etc... mediante el control exhaustivo y corrección adecuada del vector de movimiento durante el proceso de optimización.

- Además permite la determinación simultánea de un número mayor de parámetros dotados de diferentes órdenes de magnitud.

III.2.- Redes Neuronales Artificiales

Los métodos computacionales basados en la Inteligencia Artificial, han sido ampliamente utilizados en los últimos años en diversos campos de la ciencia. Entre estos métodos se encuentran los denominados Redes Neuronales Artificiales (ANN) ^(16-17, 94-102). Las Redes Neuronales son una forma de emular una de las características propias del ser humano: la capacidad de memorizar y asociar hechos. De esta manera el hombre es capaz de resolver los problemas que no pueden expresarse a través de un algoritmo mediante la experiencia acumulada y las Redes Neuronales Artificiales son sistemas que intentan reproducir esta característica humana inteligente. En definitiva, las Redes Neuronales no son más que un modelo artificial y simplificado del cerebro humano, es decir, un sistema para el tratamiento de la información cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano, la neurona.

Una Red Neuronal Artificial (ANN) es un procedimiento sistemático de procesamiento de datos, que consiste en un gran número de elementos simples y altamente interconectados mediante una arquitectura basada en la estructura del cerebro. No se basa en un modelo algebraico explícito, sino en un conjunto de unidades de activación, denominados *nodos* o *neuronas artificiales*, que están conectadas entre sí en forma de red.

III.2.1.- Sistema biológico

La neurona es la célula fundamental y básica del sistema nervioso, que es una célula alargada, especializada en conducir impulsos nerviosos. El cerebro humano contiene aproximadamente 12 billones de células nerviosas o neuronas, que están formadas por tres componentes básicos (Figura III.2.1): dendritas, cuerpo celular o soma, y axón. **Soma o cuerpo celular:** es la parte más voluminosa de la neurona, comprende una estructura esférica llamada núcleo que contiene la información que dirige la actividad de la neurona. Además en el soma se encuentra el citoplasma en el que se ubican otras estructuras que son importantes para el funcionamiento de la neurona. **Dendritas:** son prolongaciones cortas que se originan del soma, su función es recibir impulsos de otras neuronas y enviarlas hasta el soma de la neurona. **Axón:** es una prolongación única y larga, cuya función es sacar el impulso desde el soma neuronal y conducirlo hasta otro lugar del sistema. El contacto de cada axón con una dendrita se realiza a través de la *sinapsis*.

Una de las características de las neuronas es su capacidad de comunicarse. Reciben señales de entrada a través de las dendritas provenientes de estímulos externos o de otras

neuronas, el cuerpo celular las combina e integra mediante una operación generalmente no lineal y emite señales de salida que se comunican a través del axón y la sinapsis. Las señales que se utilizan son de dos tipos: eléctricas y químicas. La señal generada por la neurona y transportada a lo largo del axón es un impulso eléctrico, mientras que la señal que se transmite entre los terminales axónicos de una neurona y las dendritas de la otra es de origen químico.

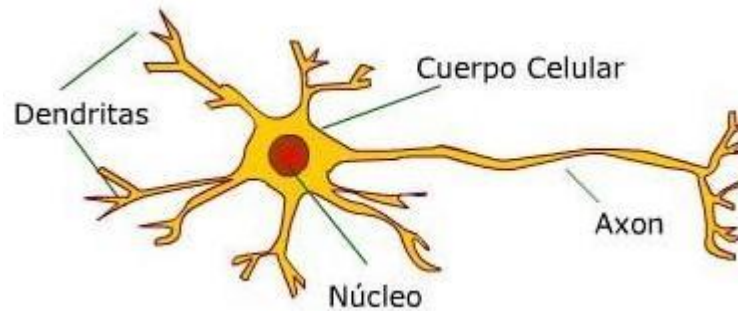


Figura III.2.1: Imagen de una neurona formada por los tres componentes básicos: soma o cuerpo celular, dendritas y axón.

III.2.2.- Redes Neuronales Artificiales

Las Redes Neuronales Artificiales, ANN (*Artificial Neural Networks*) basan su funcionamiento en las redes neuronales biológicas del cerebro humano y están constituidas por un conjunto de unidades de procesamiento, PE (*process element*), conectadas entre sí. Por analogía con el cerebro humano se denominan *neurona* o *nodo* a cada una de estas unidades de procesamiento, al igual que ocurre en las neuronas reales, cada neurona artificial recibe muchas señales de entrada y envía una única señal de salida. Las ANN al margen de *parecerse* al cerebro presentan una serie de características propias del cerebro, por ejemplo aprenden de la experiencia, generalizan a partir de ejemplos previos, ejemplos nuevos y abstraen las características principales de una serie de datos.

Tengamos en cuenta que *Aprender* es adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno, se les muestra un conjunto de entradas y ellas mismas se organizan para producir unas salidas consistentes.

Asimismo, *Generalizar* es extender o ampliar una cosa. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de “ruido”, “distorsión” o “error”.

Por su parte *Abstraer* es aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos.

Una neurona artificial o elemento procesador, PE, simula las funciones básicas de la neurona natural ⁽¹⁰³⁾ y contiene dos algoritmos, uno de los cuales calcula la suma ponderada de los valores que le llegan por las conexiones de entrada, y el otro, denominado *función de transferencia*, genera una respuesta o salida que se comunica a otras neuronas. La red de neuronas es capaz de *aprender* principalmente mediante el ajuste de los *pesos* de las conexiones entre neuronas, hasta que la red en su conjunto proporciona predicciones con suficiente precisión.

Para establecer una similitud directa entre la actividad sináptica y las Redes Neuronales Artificiales podemos considerar: las señales que llegan a la sinapsis son las entradas a la neurona, éstas son ponderadas (atenuadas o simplificadas) a través de un parámetro, denominado factor de *peso* (o simplemente *peso*) asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con *peso* positivo) o inhibirla (*peso* negativo). El efecto es la suma de las entradas ponderadas. Si la suma es igual o mayor que el umbral de la neurona, entonces la neurona se activa (da salida). La facilidad de transmisión de señales se altera mediante la actividad del sistema nervioso, las sinapsis son susceptibles a la fatiga, deficiencia de oxígeno, etc,... Esta habilidad de ajustar señales es un mecanismo de *aprendizaje*.

La Figura III.2.2 representa el elemento procesador (PE) *i*-ésimo de una Red Neuronal Artificial implementada en un ordenador. Un PE tiene varias entradas, x_i , y una sola salida x_j , que se puede aplicar a otros muchos elementos procesadores de la Red. Cada conexión con un PE determinado lleva asociada una magnitud llamada *peso* o intensidad de conexión, de este modo el *peso* de la conexión procedente del *i*-ésimo PE y que llega al *j*-ésimo, se expresa mediante w_{ji} . Las entradas se desglosan en varios tipos: si la conexión $w_{ji} > 0$ es excitatoria y si $w_{ji} < 0$ la conexión es inhibitoria. Cada PE determina un valor neto de entrada considerando todas las conexiones de entrada, que generalmente se obtiene mediante la suma ponderada de los valores de entrada, es decir, cada entrada se multiplica por su *peso* o ponderación correspondiente (w_{ji}), se suman todas las entradas ponderadas y se determina el nivel de excitación o activación de la neurona. La entrada neta de la unidad *j*-ésima se puede escribir:

$$S_j = \sum_i x_i w_{ji} \quad \text{[III.2.1]}$$

donde el subíndice *i* se extiende a todas las conexiones que posee el PE en el sumatorio.

A continuación la entrada neta, S_j , se transforma en el valor de activación, a_j , para ese PE. La activación es una función explícita de la entrada neta por lo que el valor de activación se puede escribir de la forma:

$$a_j(k) = F_i(a_j(k-1), S_j(k)) \quad \text{[III.2.2]}$$

El valor de la activación actual, $a_j(k)$, puede depender del valor anterior de la activación $a_j(k-1)$. En la mayoría de los casos, la activación y la entrada neta son idénticas, empleándose

los términos de forma intercambiable, pudiéndose utilizar la activación para denotar la entrada neta y viceversa.

Una vez que se ha calculado la activación del PE, se puede determinar el valor de salida aplicando la función de salida (*función de transferencia*):

$$x_j = f_i(a_j) \quad \text{[III.2.3]}$$

Las funciones más utilizadas son las funciones Sigmoidea y la de la Tangente hiperbólica. Puesto que normalmente $a_j = S_j$, esta función se suele escribir de la forma:

$$x_j = f_i(S_j) \quad \text{[III.2.4]}$$

El término *función de activación* se utiliza en algunas ocasiones para hacer referencia a la función f_j (*función de transferencia*), que transforma el valor de la entrada neta, S_j , en el valor de salida del nodo x_j , tal como puede verse en la Figura III.2.2.

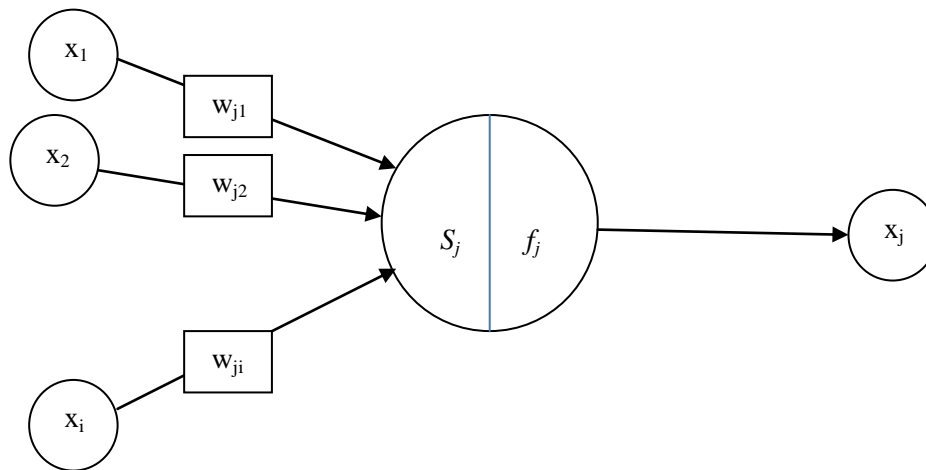


Figura III.2.2: Elemento procesador (PE) i -ésimo de una red neuronal artificial implementada en un ordenador.

III.2.3.- Características de las Redes Neuronales Artificiales

Existen cuatro aspectos que caracterizan una red neuronal: su topología (configuración adoptada), el mecanismo de *aprendizaje*, el tipo de asociación realizada entre la información de entrada y salida y la forma de representación de estas informaciones.

III.2.3.1.- Topología de las Redes Neuronales

La Topología o Arquitectura de las Redes Neuronales Artificiales consiste en la organización y disposición de las neuronas formando capas, los parámetros fundamentales de la Red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Una Red Neuronal Artificial consiste en un conjunto de unidades elementales (*neuronas, nodos o elementos procesadores*) conectadas de una forma concreta, generalmente estos elementos están organizados en grupos llamados niveles o capas. Una Red típica consiste en una secuencia de capas con conexiones entre capas adyacentes consecutivas. Existen dos

capas con conexiones con el mundo exterior, la capa de entrada (*Input layer*), donde se suministran los datos a la Red y una capa de salida (*Output layer*) que proporciona la respuesta de la Red a una entrada. El resto de las capas reciben el nombre de capas ocultas (*Hidden layers*), formadas por aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas se transfieren a neuronas de capas posteriores.

a) **Redes Monocapa:** La red más simple es un grupo de neuronas ordenadas en una única capa como se muestra en la Figura III.2.3, en la que cada una de las entradas está conectada a través de su *peso* correspondiente a la neurona artificial. En este tipo de red se establecen conexiones laterales, cruzadas o “*autorrecurrentes*” entre las neuronas que pertenecen a la única capa que constituye la red.

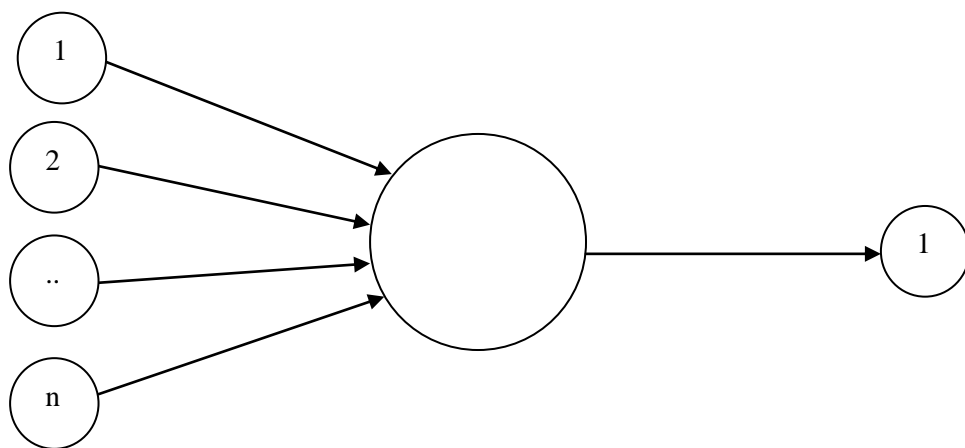


Figura III.2.3: Red Monocapa, con un grupo de neuronas ordenadas en una sola capa.

b) **Redes Multicapa:** Son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Se forman con un grupo de capas simples en cascada donde la salida de una capa es la entrada de la siguiente capa. Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior y envían señales de salida a una capa posterior. Estas conexiones se denominan *conexiones hacia delante o feedforward*. Sin embargo, en un gran número de estas redes también existe la posibilidad de conectar la salida de las neuronas de capas posteriores a la entrada de capas anteriores a estas conexiones, denominándose *conexiones hacia atrás o feedback*. Según el tipo de conexión se distinguen las Redes con conexiones hacia delante o *redes feedforward* y las Redes que disponen de conexiones tanto hacia adelante como hacia atrás o *redes feedforward/feedback*. La Figura III.2.4 muestra un ejemplo de la estructura de una Red Neuronal Artificial formada por una capa de entrada, una capa oculta y una de salida.

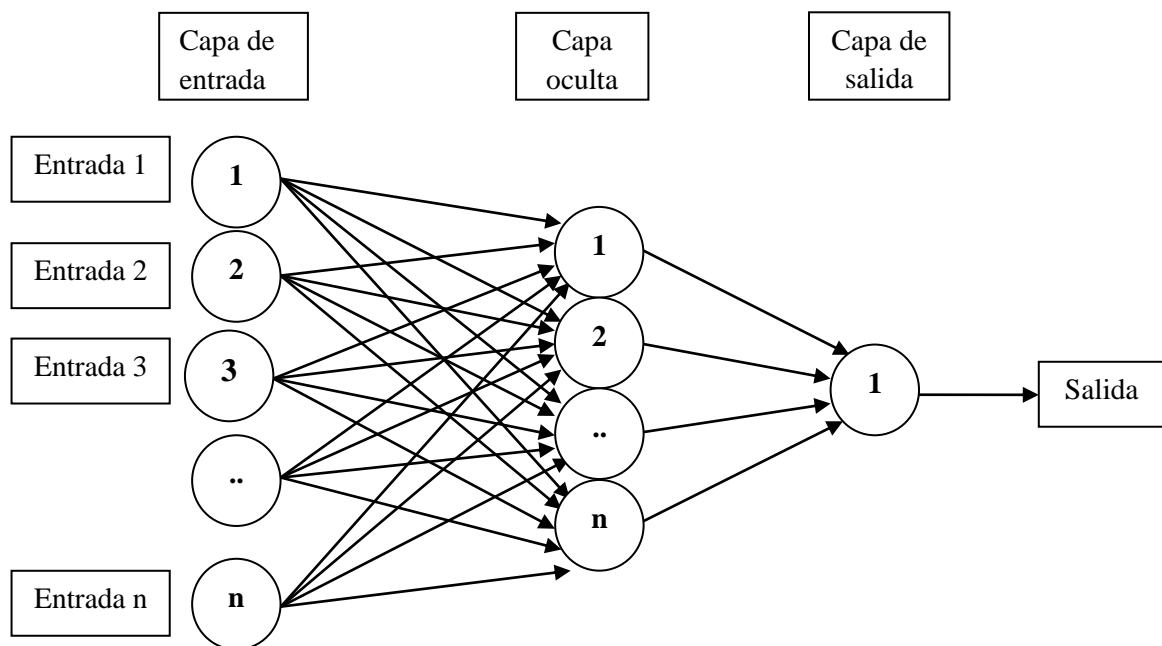


Figura III.2.4: Red Multicapa formada por una capa de entrada, una capa oculta y una de salida.

III.2.3.2.- Conexión entre neuronas

La conectividad entre los nodos (PE) de una ANN está relacionada con la forma en que las salidas de las neuronas están canalizadas para convertirse en las entradas de otras neuronas. La señal de salida de un nodo puede ser una entrada de otro elemento de proceso, o incluso ser una entrada de sí mismo (*conexión autorrecurrente*). Las neuronas pueden conectarse entre sí de diversas formas, cuando ninguna salida de las neuronas es entrada de neuronas del mismo nivel o de niveles precedentes, la Red se describe como de *propagación hacia delante (feedforward)*. Es decir, todas las señales van desde la capa de entrada hacia la salida sin existir ciclos, ni conexiones entre neuronas de la misma capa. En el caso de que las salidas puedan conectarse como entradas de neuronas de niveles previos, del mismo nivel, o incluso de ellas mismas, las Redes se denominan de *propagación hacia atrás* o *Redes recurrentes (feedforward/feedback)* y presentan al menos un ciclo cerrado de activación neuronal. Las redes de propagación hacia atrás que tienen bucles cerrados se denominan *sistemas recurrentes*.

III.2.3.3.- Mecanismo de Aprendizaje

Una Red Neuronal Artificial debe *aprender* a calcular la salida correcta para cada matriz de entrada. Este *Aprendizaje* se denomina proceso de *Entrenamiento*. El colectivo de datos (o ejemplos) sobre el cual se realiza este proceso, se llama *conjunto de datos de Entrenamiento*. La Topología de la Red y las diferentes funciones de cada neurona no pueden

modificarse durante el *Aprendizaje*, mientras que los *pesos* sobre cada una de las conexiones si pueden hacerlo ya que el *Aprendizaje* de una ANN significa: *adaptación de los pesos*. Por lo tanto, el *Aprendizaje* es el proceso por el cual una Red Neuronal Artificial modifica sus *pesos* en respuesta a una información de entrada. Los cambios que se producen durante el mismo se reducen a la desaparición, modificación y creación de conexiones entre las neuronas. La creación de una nueva conexión implica que el *peso* de la misma pasa a tener un valor distinto de cero, de la misma manera, una conexión desaparece cuando su *peso* pasa a ser cero. Durante el proceso de *Aprendizaje*, los *pesos* de las conexiones de la Red sufren modificaciones y el proceso finaliza (la Red ha *aprendido*) cuando los valores de los *pesos* permanecen constantes, es decir cuando $dw_{ji}/dt = 0$. La modificación en los *pesos* (w_{ji}) se produce de la siguiente manera:

$$w^{nuevo} = w^{viejo} + \Delta w \quad \text{[III.2.5]}$$

Este cambio puede expresarse matemáticamente para una determinada etapa de *Aprendizaje* como:

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \quad \text{[III.2.6]}$$

donde k se refiere a la etapa de *Aprendizaje*, $w_{ji}(k+1)$ al nuevo valor del *peso* y $w_{ji}(k)$ al valor anterior.

En definitiva una de las principales características de las ANN es su capacidad de *Aprendizaje (Learning)*, para lo cual se lleva a cabo el *Entrenamiento (Training)* de la Red de una manera que posee algunos paralelismos con el desarrollo intelectual de los seres humanos. El objetivo del *Entrenamiento* de una ANN es conseguir que una aplicación determinada produzca el conjunto de salidas deseadas o correctas para un conjunto de entradas. El proceso consiste en la aplicación secuencial de diferentes conjuntos de entradas para que se ajusten los *pesos* de las interconexiones según un procedimiento predeterminado. Durante el *Entrenamiento* los *pesos* convergen gradualmente hacia los valores que hacen que cada entrada produzca la salida deseada.

Los métodos de *Entrenamiento* o los procedimientos de ajuste de los valores de las conexiones de las ANN se pueden clasificar en dos grupos: Supervisado y No Supervisado. Esto da lugar a una de las clasificaciones más importantes que se realizan de las ANN: Redes Neuronales Artificiales con *Aprendizaje Supervisado* y Redes Neuronales Artificiales con *Aprendizaje No Supervisado*. La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo que controle el *Aprendizaje* de la Red.

a) Aprendizaje Supervisado.

El proceso de *Aprendizaje* se realiza mediante un *Entrenamiento* controlado por un agente externo (Supervisor) que determina la respuesta que debería generar la Red a partir de una entrada determinada. El supervisor controla la salida de la Red y en caso de que ésta no

coincida significativamente con la correcta, se procederá a modificar los *pesos* de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada. Los algoritmos de *Aprendizaje Supervisado* requieren la comparación de cada matriz de entrada con su correspondiente matriz de salida. El *Entrenamiento* consiste en suministrar una matriz de entrada a la Red, calcular la salida de la misma, compararla con la matriz correcta y el error o diferencia resultante se utiliza para realimentar la Red tras modificar los *pesos* de acuerdo con un algoritmo que tiende a minimizar el error. Las parejas de matrices del *conjunto de Entrenamiento* se aplican secuencialmente de forma cíclica mediante un proceso iterativo, se calcula el error y el ajuste de los *pesos* por cada pareja, hasta que el error para el proceso global de *Entrenamiento*, alcance un valor suficientemente pequeño y aceptable. Este tipo de *Aprendizaje Supervisado* se puede llevar a cabo de diferentes formas, que dan lugar a diferentes tipos:

a1) Aprendizaje por corrección del error. Consiste en ajustar los *pesos* de las conexiones de la Red en función de la diferencia entre los valores correctos y los obtenidos a la salida de la Red, es decir, en función del error cometido en la salida. Un ejemplo de este tipo de algoritmos lo constituye la *regla de Aprendizaje del Perceptron*, utilizada en el *Entrenamiento* de la Red del mismo nombre. Se basa en la aplicación de la *regla de Aprendizaje Delta* o regla del mínimo error cuadrático (LMS Error: *Least Mean Squared Error*). También pertenece a esta clasificación la *regla de aprendizaje de propagación hacia atrás o de Backpropagation*, regla LMS multicapa, que es una generalización de la *regla de Aprendizaje Delta*.

a2) Aprendizaje por refuerzo. Es más lento que el anterior, se basa en la idea de no disponer de un ejemplo completo del comportamiento deseado, es decir, de no indicar durante el *Entrenamiento* exactamente la salida que se desea que proporcione la Red ante una determinada entrada. En el *Aprendizaje por refuerzo* la función del supervisor se reduce a indicar mediante una señal de *refuerzo* si la salida obtenida en la Red se ajusta o no a la deseada (éxito = +1 o fracaso = -1), y en función de ello se ajustan los *pesos* basándose en un cálculo de probabilidades.

a3) Aprendizaje estocástico. Consiste básicamente en realizar cambios aleatorios en los valores de los *pesos* de las conexiones de la Red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

b) Aprendizaje No Supervisado.

El proceso de *Aprendizaje* no requiere influencia externa para ajustar los *pesos* de las conexiones entre neuronas. La Red no recibe ninguna información por parte del entorno que le indique si la salida generada es o no correcta. Estos métodos fueron desarrollados por T. Kohonen ⁽⁹⁷⁻⁹⁸⁾ y no requieren de una matriz de salida deseada y por tanto no se realizan comparaciones entre las salidas reales y salidas correctas. El conjunto de matrices de *Entrenamiento* consiste únicamente en matrices de entrada y el algoritmo de *Entrenamiento*

modifica los *pesos* de la Red de forma que produzca matrices de salida autoconsistentes. El proceso de *Entrenamiento* da lugar a las propiedades estadísticas del conjunto de matrices de *Entrenamiento* y agrupa en clases las matrices similares. Existe una gran variedad de algoritmos de *Entrenamiento*, la mayoría de ellos han surgido de la evolución del modelo de *Aprendizaje* no Supervisado que propuso Hebb ⁽¹⁰⁴⁾. En general en este tipo de *aprendizaje* se suelen considerar dos tipos: *Aprendizaje* Hebbiano y *Aprendizaje* competitivo y comparativo.

b1) *Aprendizaje* Hebbiano. Esta regla de *Aprendizaje* es la base de muchas otras que pretende extraer características de los datos de entrada y se fundamenta en la siguiente suposición: si dos neuronas N_i y N_j tienen simultáneamente el mismo estado (ambas activas o ambas inactivas), el *peso* de la conexión entre ambas se incrementa. Las entradas y salidas permitidas a la neurona son: $\{-1, 1\}$ o $\{0, 1\}$ (neuronas binarias). Esto se explica porque la regla de aprendizaje de Hebb se basa en la neurona biológica clásica, que solamente puede tener dos estados, activa o inactiva.

b2) *Aprendizaje* competitivo y comparativo. Este tipo de *Aprendizaje* está orientado a la clasificación de los datos de entrada. Puede ocurrir que un patrón nuevo pertenezca a una clase que sea reconocida por la Red con lo que se produce la inclusión de este nuevo patrón en esta clase. Si el patrón no pertenece a ninguna de las clases reconocidas anteriormente, entonces se ajustan la estructura y los *pesos* de la Red Neuronal para reconocer la nueva clase.

III.2.3.4.- Tipo de Asociación entre las Informaciones de Entrada y Salida.

Las ANN son sistemas que almacenan cierta información *aprendida*. Esta información se distribuye en los *pesos* asociados a las conexiones entre neuronas de entrada y salida. Existen dos formas primarias de realizar esa asociación de entrada/salida: *heteroasociación* y *autoasociación*.

a) *Heteroasociación*. En este tipo de asociación la Red *aprende* parejas de datos $[(A_1, B_1), (A_2, B_2) \dots (A_n, B_n)]$, de tal forma que cuando se presente cierta información de entrada A_i , la Red deberá responder generando la correspondiente salida B_j . Una Red *heteroasociativa* computa cierta función, que en la mayoría de los casos no puede expresarse analíticamente, entre un conjunto de entradas y un conjunto de salidas, correspondiendo a cada posible entrada una determinada salida. Existen Redes *heteroasociativas* con conexiones *feedforward*, *feedforward/feedback* y con conexiones laterales. Su *Aprendizaje* puede ser supervisado o no supervisado.

b) *Autoasociación*. En este caso la Red *aprende* ciertas informaciones $A_1, A_2 \dots A_n$, de tal forma que cuando se le suministra una información de entrada realizará una *autocorrelación*, respondiendo con uno de los datos almacenados, eligiendo el más similar al de la entrada. La Red *autoasociativa* reconstruye una determinada información de entrada que se presenta incompleta o distorsionada (le asocia el dato almacenado más parecido). Pueden implementarse

en una Red constituida por una sola capa. Posee conexiones laterales o también *autorrecurrentes* y habitualmente son de *Aprendizaje* no supervisado.

III.2.4.- Tipos de Redes Neuronales Artificiales.

III.2.4.1.- *Perceptrón*

Tal y como hemos señalado anteriormente una de las características más significativas de las Redes Neuronales Artificiales es su capacidad para *aprender* a partir de alguna fuente de información interactuando con su entorno. F. Ronsenblat ⁽¹⁰⁵⁾ desarrolló un modelo simple de neurona, llamado *Perceptrón*, basado en el modelo de McCulloch y Pitts ⁽¹⁰⁶⁾ y en una regla de *Aprendizaje* basada en la corrección del error. El *Perceptrón* simple ⁽¹⁰⁵⁾ presenta dos capas de unidades procesadoras (neuronas) y sólo una de ellas presenta la capacidad de adaptar o modificar los *pesos* de las conexiones, utiliza un algoritmo de *Aprendizaje* supervisado. La Figura III.2.5 muestra la unidad procesadora básica del *Perceptrón*. Las entradas x_i proceden de la parte izquierda y cada conexión con la neurona j tiene asignada un *peso* de valor w_{ji} .

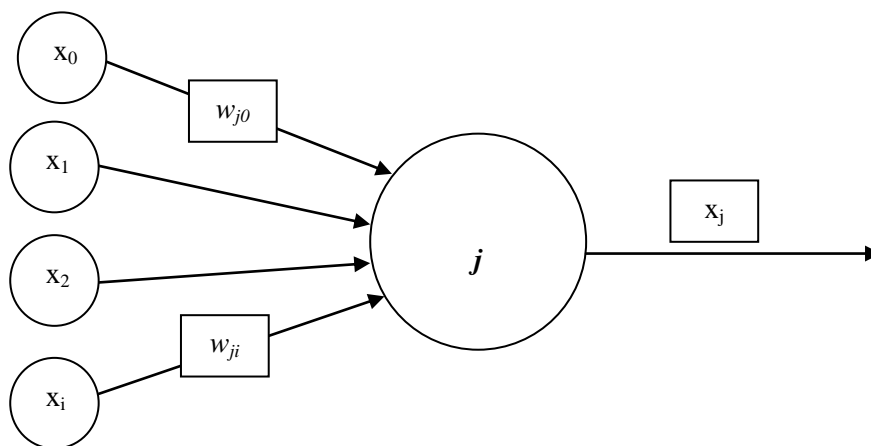


Figura III.2.5: Unidad procesadora básica del *Perceptrón*.

La unidad procesadora del *Perceptrón* realiza la suma ponderada de las entradas de acuerdo con la ecuación III.2.7.

$$S_j = \sum x_i w_{ji} \quad \text{[III.2.7]}$$

Un aspecto común en muchas ANN es la entrada especial llamada *bias* (entrada x_0) que siempre presenta un valor fijo +1. El *Perceptrón* comprueba si la suma de las entradas ponderadas es mayor o menor que un cierto valor umbral o sesgo (θ) y genera la salida x_j de acuerdo con:

$$\text{si } S_j > \theta \text{ entonces } x_j = 1$$

$$\text{si } S_j \leq \Theta \text{ entonces } x_j = 0 \quad \text{[III.2.8]}$$

La salida x_j se transmite a lo largo de la línea de salida y constituye uno de los componentes de la matriz de salida de la Red.

El *Entrenamiento* del *Perceptrón* consiste en suministrar a la Red todos los elementos del proceso global de *Entrenamiento* constituido por parejas de matrices (entrada y salida deseada) de forma secuencial. El objetivo del *Entrenamiento* es llegar a un conjunto de valores de los *pesos* de la Red de forma que responda correctamente al proceso global de *Entrenamiento*. Después del *Entrenamiento* los *pesos* no son modificados y la Red está en disposición de responder adecuadamente a las entradas que se le suministren. La adaptación de los *pesos* se puede realizar mediante diferentes reglas, una de las reglas más simples de *Aprendizaje* del *Perceptrón* es la siguiente:

$$w_{ji}^{posterior} = w_{ji}^{anterior} + C(t_j \times x_j) x_i \quad \text{[III.2.9]}$$

t_j es el valor de la salida deseada, x_j el valor de salida producida por la unidad procesadora, x_i el valor de la entrada i y C el coeficiente de *Aprendizaje*.

En todo proceso de *Entrenamiento* el comportamiento de la Red va mejorando hasta que llega a un punto en el que se estabiliza y se dice que la Red converge. Esta convergencia tiene dos posibilidades, la primera consiste en que la Red haya *aprendido* correctamente del conjunto de *Entrenamiento* y la segunda se trata de que la Red no haya *aprendido* todas las respuestas correctas. Las redes *Perceptrón* de dos capas tienen una capa de entrada y una capa de unidades procesadoras que constituyen la capa de salida. Existen muchos tipos de topologías de Redes basadas en la arquitectura del *Perceptrón*, las Redes *Perceptrón* de dos capas tienen una capa de entrada y una capa de unidades procesadoras que constituyen la capa de salida. Las topologías con tres o más capas se caracterizan porque la regla de *Aprendizaje* sólo influye sobre los *pesos* de una capa.

III.2.4.2.- *Adaline* y *Madaline*

La Arquitectura de *Adaline* (*Adaptive Linear Neuron*), creada por B. Widrow ⁽¹⁰⁷⁻¹⁰⁸⁾, utiliza un dispositivo lógico que realiza una suma lineal de las entradas y genera una función umbral para el resultado de dicha suma. En una red *Adaline* básica la unidad procesadora implementa una función umbral y las conexiones de cada una de las entradas tienen asociadas un valor de ponderación llamado también *peso* w_i . El mecanismo de ajuste de los *pesos* utiliza la diferencia entre el valor de la salida y el valor esperado, la unidad procesadora actúa como un sumatorio y después realiza la función umbral de acuerdo con;

$$\begin{aligned} x_j &= 1 \text{ si } S = \sum_i x_i w_i \geq 0 \\ x_j &= -1 \text{ si } S = \sum_i x_i w_i < 0 \end{aligned} \quad \text{[III.2.10]}$$

La salida de la unidad *Adaline* es ± 1 a diferencia de la arquitectura del *Perceptrón* que sólo permite los valores 0 y 1.

El *Entrenamiento* se realiza suministrando repetidamente una serie de parejas de entradas y salidas. El objetivo de la *Adaline* durante el proceso de la adaptación es producir la salida deseada como suya propia. La regla de *Aprendizaje* en la arquitectura de la *Adaline* es la regla de Widrow-Hoff⁽¹⁰⁷⁾ expresada en la ecuación:

$$\Delta w_i = \eta x_i (t - x) \quad \text{[III.2.11]}$$

donde η es la constante de *Aprendizaje*, x_i la salida de la unidad i , t la salida deseada y por último x la salida de la unidad *Adaline*, un modificación de esta regla considera el valor de la suma ponderada S en lugar del valor de la salida.

La arquitectura ***Madaline*** (*Multilayer Adaline*) diseñada también por Widrow⁽¹⁰⁷⁻¹⁰⁸⁾, presenta una configuración constituida por dos o más unidades *Adaline*. El sistema tiene una capa de unidades *Adaline* que están conectadas a una simple unidad *Madaline*. Las conexiones entre la capa de entrada y la capa de las unidades *Adaline* llevan asociadas valores de *peso* ajustables por cada una de ellas. Sin embargo, las conexiones entre la capa de *Adaline* y la unidad *Madaline*, no llevan asociado ningún *peso*. Cada unidad *Adaline* transmite su salida (-1 ó $+1$) a la unidad *Madaline* que utiliza el criterio de la mayoría para decidir la salida. Así, si la mitad o más de las unidades *Adaline* poseen un valor de salida $+1$, entonces la salida de la *Madaline* es $+1$; en caso contrario el valor de salida de la red *Madaline* es -1 . El *Entrenamiento* de los sistemas *Madaline* es similar al de las *Adaline*. Se utiliza un proceso de calibrado con patrones estándar cuyas respuestas son perfectamente conocidas “*a priori*”. Una vez que se suministra el patrón a la entrada, el sistema *Madaline* calcula su salida y a continuación se compara con la salida correcta. Los *pesos* se modifican “*a posteriori*”.

III.2.4.3.- Red *Backpropagation*

El desarrollo del algoritmo *Backpropagation*⁽¹⁰⁹⁻¹¹⁰⁾ ha desempeñado un papel muy importante en la aplicación de las Redes Neuronales Artificiales. Se trata de un método de *Entrenamiento* de redes multicapa cuya importancia reside en su capacidad de entrenar las capas ocultas. De este modo, se incrementan las posibilidades más reducidas de las redes de una única capa.

a) Arquitectura de la Red *Backpropagation*

La unidad procesadora básica se encuentra en la zona central situándose las entradas en su parte izquierda. Las unidades que reciben las salidas se encuentran en la parte derecha tal como puede apreciarse en la Figura III.2.6. La unidad procesadora tal y como hemos visto anteriormente se caracteriza por realizar una suma ponderada de las entradas, S_j , producir una salida x_j y poseer un valor δ_j asociado que se utilizará en el proceso de ajuste de los *pesos*. El

peso asociado a la conexión desde la unidad i a la unidad j se representa por w_{ji} y puede ser modificado durante el proceso global de *Aprendizaje*.

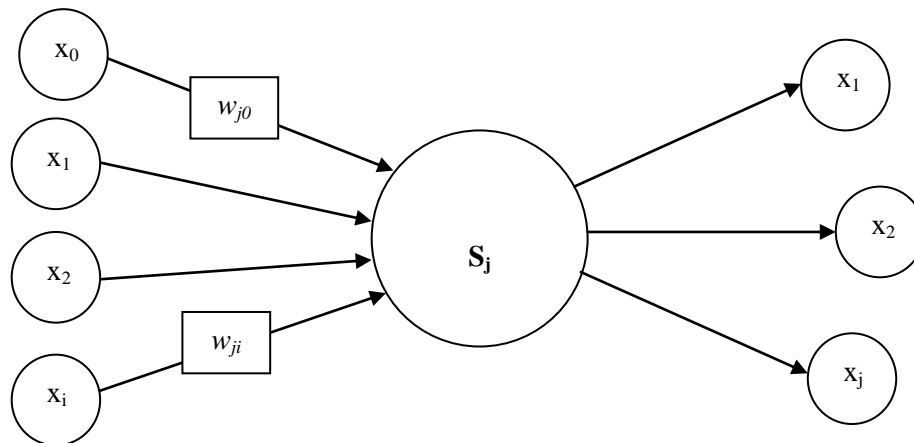


Figura III.2.6: Unidad procesadora básica de la Red *Backpropagation*.

La Red *Backpropagation* utiliza normalmente tres o más capas de unidades procesadoras. La capa inferior es la capa de entrada (*Input*). Se caracteriza por ser la única capa cuyas unidades procesadoras reciben entradas desde el exterior. Se trata de puntos distribuidores que no realizan ninguna operación de cálculo, mientras las unidades procesadoras de las demás capas procesan las señales. La siguiente capa es la capa oculta (*Hidden*) y todas sus unidades procesadoras están interconectadas con la capa inferior y con la capa superior que es la capa de salida (*Output*) que produce la respuesta de la Red. Pueden existir varias capas ocultas (*Hidden*) interconectadas entre sí y con las capas de entrada (*Inputs*) y salida (*Outputs*). El establecimiento de la arquitectura de la Red neuronal, es una cuestión crucial para conseguir los mejores resultados.

b) Método de Entrenamiento

Las Redes *Backpropagation* tienen un método de *Entrenamiento* supervisado. La Red aprende un conjunto definido “*a priori*” de pares de entradas y salidas (*patrones*), propuestos como ejemplos, empleando un ciclo de *propagación-adaptación* de dos fases. A la Red se le proporcionan parejas de patrones (un patrón de entrada se empareja con un patrón de salida deseada y los *pesos* se ajustan de forma que disminuya el error entre la salida correcta) y la respuesta de la Red. El algoritmo de *Aprendizaje Backpropagation* conlleva una fase de propagación hacia delante (*forward*) y otra fase de propagación hacia atrás. Ambas fases se realizan por cada patrón proporcionado en la sesión de *Entrenamiento*.

b1) Propagación hacia delante (*forward*).

Esta fase de propagación hacia delante se inicia cuando se presenta un patrón en la capa de entrada de la Red. Cada unidad de la entrada se corresponde con un elemento del vector patrón de entrada. Las unidades de entrada toman el valor del elemento de la matriz patrón de

entrada y se calcula el valor de activación o nivel de salida de la primera capa. Posteriormente las demás capas realizarán la fase de propagación hacia delante que determina el nivel de activación de las otras capas. La unidad procesadora j calcula la suma ponderada S_j de acuerdo con la ecuación:

$$S_j = \sum_i x_i w_{ji} \quad \text{[III.2.12]}$$

y genera la salida, x_j , o nivel de activación:

$$\text{Salida} = x_j = f_j(S_j) \quad \text{[III.2.13]}$$

La función f_j (*función de transferencia*), es una función genérica umbral. Entre las más utilizadas, se encuentran la función Sigmoidea y la función Hiperbólica. El valor de la salida de la unidad j es enviado o transmitido a lo largo de todas las conexiones de salida de dicha unidad.

b2) Propagación hacia atrás.

Una vez se ha completado la fase de propagación hacia delante, se inicia la fase de corrección o fase de propagación hacia atrás. Las modificaciones de todos los *pesos* de las conexiones empiezan por la capa de salida y continúan hacia atrás, a través de todas las capas de la Red, hasta la capa de entrada. Los tipos de ajuste de los *pesos* se pueden clasificar en dos grupos: ajuste de unidades procesadoras de la capa de salida y ajuste de unidades procesadoras de las capas ocultas.

- *Ajuste de Pesos de la Capa de Salida:* el ajuste de estos *pesos* es relativamente sencillo debido a que se conoce el valor deseado para cada una de las unidades de la capa de salida. Cada unidad de la capa de salida produce un número real como salida y se compara con el valor deseado, el cual está especificado en el patrón a suministrar en el *Entrenamiento*. A partir del resultado de la comparación, se calcula un valor de error δ_j para cada unidad de la capa de salida mediante la ecuación III.2.14:

$$\delta_j = (t_j - x_j) f_j'(S_j) \quad \text{[III.2.14]}$$

donde t_j es el valor de salida deseado y f_j' la derivada de la función f_j .

- *Ajuste de Pesos de las Capas Ocultas:* estas capas no tienen una matriz de salida con valores correctos y por tanto no se puede seguir el método de propagación de error mencionado en el caso de unidades procesadoras de la capa de salida. El valor de error, calculado para este tipo de unidades procesadoras, se obtiene a partir de la ecuación III.2.15:

$$\delta_j = [\sum_k \delta_k w_{jk}] f_j'(S_j) \quad \text{[III.2.15]}$$

Los *pesos* (w_{ji}) asociados a las conexiones sufren modificaciones, para una determinada etapa k de aprendizaje:

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \quad \text{[III.2.16]}$$

donde $w_{ji}(k+1)$ se refiere al valor posterior del peso y $w_{ji}(k)$ al del anterior.

El ajuste de los *pesos* asociados a las conexiones se lleva a cabo a partir del valor δ de cada unidad de proceso, mediante la aplicación de la regla Delta generalizada ⁽¹⁰⁹⁾:

$$\Delta w_{ji} = \eta \delta_j x_i \quad \text{[III.2.17]}$$

La variable η es el coeficiente de *Aprendizaje* que normalmente toma valores entre 0.25 y 0.75 y refleja el grado de *Aprendizaje* de la Red. En ocasiones este coeficiente se modifica desde un valor inicial alto a valores progresivamente menores durante el proceso de *Entrenamiento* con el objetivo de lograr un mejor *Aprendizaje*.

- *Convergencia*: en el proceso de *Entrenamiento* o *Aprendizaje* de la Red *Backpropagation* es frecuente medir cuantitativamente el *Aprendizaje* mediante el valor RMS (*Root Mean Square*) del error de la Red. Esta medida refleja el modo en el que la Red logra respuestas correctas. A medida que la Red *aprende* el valor RMS decrece. Debido a que los valores de salida de la Red y los valores de salidas deseadas son valores reales, es necesario definir un valor umbral del valor RMS que permita decidir cuando la Red se aproxima a la salida deseada en orden a considerar que la respuesta es correcta. La convergencia es un proceso en el que el valor RMS del error de la Red tiende progresivamente al valor 0. No siempre es fácil conseguir la convergencia porque a veces el proceso es muy lento y requiere un tiempo excesivo para alcanzarla. También puede ocurrir que la Red alcanza un mínimo local y en ese momento, deja de *aprender*.

III.2.4.4.- Red *Self Organizing Map*

La Red *Self Organizing Map* (SOM) presenta una topología constituida por dos capas. La primera capa es la de *entrada* y la segunda, llamada capa competitiva o de T. Kohonen, está organizada en una rejilla de dos dimensiones. Las dos capas están totalmente interconectadas, cada una de las conexiones lleva asociado un *peso* que será modificado a lo largo de la sesión de *Entrenamiento*. Asimismo, utiliza un mecanismo de *Aprendizaje* no supervisado.

III.2.4.5.- Red *Counterpropagation*

La Red *Counterpropagation* fue desarrollada por R. Hecht-Wielsen ⁽⁹⁹⁾ y combina varias capas de distintas Redes o Arquitecturas diferentes para la construcción de un nuevo tipo de Red. Se utilizan dos tipos diferentes de capas: la capa oculta es una capa de T. Kohonen con neuronas competitivas y *Aprendizaje* no supervisado. La capa de salida que está totalmente conectada a la capa oculta, es no competitiva. Una topología o Arquitectura típica de la Red *Counterpropagation* está formada por tres capas, la primera capa la constituye la capa de entrada, la segunda capa es la capa competitiva o de T. Kohonen y la tercera capa es la capa de salida llamada también capa de Grossberg. Durante el *Entrenamiento* se ajustan los *pesos* de las conexiones de las dos capas, en primer lugar, los correspondientes a la capa de T. Kohonen y después los *pesos* de la capa de salida. Sólo se ajustan los *pesos* de las conexiones entre la entrada y la neurona ganadora, permaneciendo los demás *pesos* inalterados.

III.2.4.6.- Red *Hopfield*

La red *Hopfield* ⁽¹¹¹⁾ tiene una única capa de unidades procesadoras. Cada una de las unidades procesadoras tiene un valor o nivel de activación (también llamado estado) que es de tipo binario [0/1]. La red *Hopfield* tiene un estado en cada momento. Este estado se define por el vector de unos y ceros constituido por los estados de todas las unidades procesadoras. Las unidades procesadoras de la red *Hopfield* están completamente interconectadas. Esta topología convierte a la red *Hopfield* en una red recursiva ya que la salida de cada unidad está realimentada con las entradas de las demás unidades. Una característica de las redes *Hopfield* es la doble conexión por cada pareja de unidades procesadoras. Además los *pesos* asignados a ambas conexiones tienen el mismo valor.

III.2.4.7.- Red *Bidirectional Associative Memory*

El modelo de la Red BAM (*Bidirectional Associative Memory*), desarrollada por Kosko ^(103,112), es una Red *heteroasociativa* ya que acepta una matriz de entrada en un conjunto de neuronas y produce otra matriz relacionada, o matriz de salida, en otro conjunto de neuronas. La Red tiene dos capas centrales de neuronas totalmente interconectadas además de las capas de entrada y salida. La Red se diseña para almacenar parejas asociadas de matrices y los *pesos* de las conexiones entre las capas centrales almacenan la información asociativa. Si la primera capa tiene N neuronas y la segunda capa tiene M neuronas los *pesos* de estas conexiones se almacenan en una matriz W de dimensiones NxM. En esta arquitectura se aplica un vector de entrada, A, a los *pesos* W (es decir a la salida de la primera capa) y se produce un vector de salida B. Cuando se aplica el vector B a la matriz traspuesta W^T se producen nuevas salidas para el vector A. Este proceso se repite hasta que la red alcanza un punto estable en el que A y B no cambian.

III.2.4.8.- Red *Adaptive Resonance Theory*

La Red *Adaptive Resonance Theory* (ART) tiene sus orígenes en las ideas de Grossberg ⁽¹¹³⁻¹¹⁴⁾ y presenta dos modelos, ART1 que acepta solo matrices de entradas binarias (base 2, [0/1]) (*digitales*) y ART2 que admite también entradas continuas (*analógicas*). El núcleo de la Red ART consiste en dos capas interconectadas y en una serie de bloques que realizan funciones de control requeridas en las fases de *Entrenamiento* y clasificación. Esta red es un clasificador de vectores. Admite vectores de entrada y los clasifica en una de las categorías posibles en función del patrón que tiene almacenado al que más se aproxime. La capa de reconocimiento es la responsable de indicar la categoría. En el caso de que el vector de entrada no se aproxime suficientemente a ningún patrón almacenado se crea una nueva categoría almacenando un patrón idéntico a la entrada. Una vez encontrado un patrón que se

parezca al vector de entrada que satisface una tolerancia especificada (parámetro de *vigilancia*), se ajusta o se *entrena* dicho patrón para aproximarlo aún más al vector de entrada.

III.3.- Optimización Matemática. Aspectos Teóricos del Algoritmo Generalizado de Descenso Controlado

Uno de los objetivos del trabajo que recoge la presente Memoria es la determinación de Parámetros Cinéticos y Termodinámicos de Activación para distintos mecanismos de reacción mediante un proceso de optimización matemática tomando como estimaciones iniciales los valores de los parámetros obtenidos tras la aplicación de ANN. Hemos utilizado el Algoritmo de optimización matemática AGDC (Algoritmo General de Descenso Controlado) ^(68-69,72). Este algoritmo ha sido diseñado, puesto a punto y exhaustivamente probado en nuestro Departamento, avalado por su amplia utilización en una extensa casuística de sistemas químico físicos ^(65-74,115-131).

La optimización matemática puede ser una minimización o maximización. En nuestro caso pretendemos minimizar la función suma de desviaciones cuadráticas, SQD, cuyas variables independientes son los valores de los parámetros que se pretenden determinar. El procedimiento propuesto es numérico, por tanto, partiendo de unas estimaciones iniciales de los parámetros y mediante el proceso de optimización, se calculan los valores que hacen mínimo el valor de la función suma de desviaciones cuadráticas (SQD).

Los métodos de optimización matemática utilizados para la determinación de los diferentes parámetros son métodos aproximados iterativos, dentro de los cuales existen dos tipos que se diferencian por su fundamento matemático, los métodos de búsqueda y los de gradiente. El algoritmo AGDC es un método de optimización de gradiente, por lo que explicaremos las características matemáticas fundamentales de los métodos de gradiente y en concreto del algoritmo AGDC. También explicaremos los fundamentos matemáticos de las distintas partes que son necesarias para llevar a cabo el proceso global.

III.3.1.- Aspectos generales de los métodos de gradiente

Antes de realizar una exposición del algoritmo de optimización AGDC, vamos a llevar a cabo un estudio matemático general de las características más importantes en las que se basa el algoritmo.

Hemos utilizado la notación “*bold type*” para representar los vectores y/o matrices de las ecuaciones correspondientes.

Consideremos la función $Y: R^1 \times R^n \rightarrow R^l$ que depende de “ l ” variables independientes (vector \mathbf{t}) y de n parámetros (vector \mathbf{X}):

$$y = Y(\mathbf{t}, \mathbf{X}) \quad \mathbf{t} \in R^l, \mathbf{X} \in R^n \quad \text{[III.3.1]}$$

Si disponemos de N_d datos experimentales de cada variable independiente \mathbf{t}_i y de la función \mathbf{y}_i , se podrá realizar un ajuste para la obtención de un valor \mathbf{X}^* de \mathbf{X} que verifique:

$$y_i = Y(\mathbf{t}_i, \mathbf{X}^*) \quad \forall i(i = 1, 2, \dots, N_d) \quad \text{[III.3.2]}$$

Si el vector \mathbf{X}^* satisface la igualdad anterior, será el minimizador de una función SQD: $R^n \rightarrow R^l$ denominada función suma de desviaciones cuadráticas y que se define:

$$SQD(\mathbf{X}) = \sum_{i=1}^{N_d} (Y(\mathbf{t}_i, \mathbf{X}) - y_i)^2 \quad i = (1, 2 \dots N_d) \quad \mathbf{X} \in R^n \quad \text{[III.3.3]}$$

Si la función Y es lineal en x_1, x_2, \dots, x_n , la resolución de un sistema de N_d ecuaciones con n incógnitas posee una solución exacta. Para determinar la solución de este sistema se pueden aplicar alguno de los algoritmos clásicos desarrollados para este propósito. Si la función Y no es lineal hay que utilizar métodos numéricos de optimización apropiados para llevar a cabo la resolución del sistema.

El problema fundamental de la optimización matemática consiste en minimizar la cantidad escalar E , que es el valor de una función SQD (suma de desviaciones cuadráticas) de N_p parámetros del sistema: X_1, X_2, \dots, X_p . Estos parámetros deben ser ajustados para obtener el mínimo deseado, es decir, minimizar $E = SQD(X_1, X_2, \dots, X_p)$. Con todo esto, formulamos el problema de la optimización como un problema de minimización, pero esto no es restrictivo, ya que muchos problemas pueden generalizarse de esta forma ⁽¹³²⁾.

Adoptando una nueva nomenclatura, la función suma de desviaciones cuadráticas adoptará la siguiente forma:

$$SQD = \sum_{i=1}^{N_d} (f_i(X_1, X_2, \dots, X_p, t_i) - (y_i)_E)^2 \quad \text{[III.3.4]}$$

donde f_i representa la función analítica que proporciona el valor teórico de la propiedad (concentración, absorbancia,..) con la que se sigue la evolución del sistema en estudio, X_1, X_2, \dots, X_p son los parámetros que se pretenden determinar e $(y_i)_E$ es el valor experimental de la propiedad obtenida a los distintos valores de tiempo.

Los N_p parámetros del sistema constituyen el vector columna \mathbf{X} y su transpuesto puede expresarse, en forma matricial, como:

$$\mathbf{X}^T = [X_1 X_2 \dots X_p] \quad \text{[III.3.5]}$$

Las coordenadas de \mathbf{X} cambiarán sus valores a medida que se realice un ajuste de los parámetros durante el proceso de optimización. Se denomina iteración m a cada serie de ajustes de estas variables. En general, son necesarias varias iteraciones antes de aproximarse satisfactoriamente al mínimo. Debe introducirse una primera estimación de los valores de los parámetros $\mathbf{X}^{(0)}$ como punto de partida en el proceso iterativo para la búsqueda del valor mínimo de la función SQD , con el fin de obtener el primer valor de la función objetivo que será

$SQD^{(0)}$. Al final de la optimización se obtendrán unos valores óptimos que se suelen denotar como \mathbf{X}^* y SQD^* .

El punto \mathbf{X}^* , que da el menor valor de SQD , se denomina minimizador global y SQD^* es el menor valor posible de SQD .

La variación de los valores de los parámetros se agrupa en el vector $\Delta\mathbf{X}$, al que denominamos **vector de movimiento**:

$$\Delta\mathbf{X}^T = [\Delta X_1 \Delta X_2 \dots \dots \Delta X_p] \quad \text{[III.3.6]}$$

En la práctica, es muy difícil determinar si el minimizador obtenido por un proceso numérico es un minimizador global. En la mayoría de los casos podemos decir que es un minimizador en el área local de búsqueda y por ello a \mathbf{X}^* se le denomina minimizador local. Una función puede poseer varios minimizadores locales.

DEFINICIONES:

Consideremos una función $S: R^n \longrightarrow R^l$ y un conjunto D de R^n :

Se dice que $\mathbf{X}^* \in D$ es un **fuerte minimizador** local de S en D si y sólo si existe un $\varepsilon > 0$ tal que $B(\mathbf{X}^*, \varepsilon) \subset D$ y $S(\mathbf{X}) > S(\mathbf{X}^*)$, $\forall \mathbf{X} \in B(\mathbf{X}^*, \varepsilon)$, $\mathbf{X} \neq \mathbf{X}^*$

Se dice que $\mathbf{X}^* \in D$ es un **débil minimizador** local de S en D si y sólo si existe un $\varepsilon > 0$ tal que $B(\mathbf{X}^*, \varepsilon) \subset D$ y $S(\mathbf{X}) \geq S(\mathbf{X}^*)$, $\forall \mathbf{X} \in B(\mathbf{X}^*, \varepsilon)$

Una matriz simétrica \mathbf{A} se dice que es **definida positiva** si todos los valores propios de dicha matriz son estrictamente positivos.

Si todos los valores propios de la matriz simétrica \mathbf{A} son mayores o iguales a cero, se dice que \mathbf{A} es **semidefinida positiva**.

Si la matriz simétrica \mathbf{A} tiene valores positivos y negativos, se dice que \mathbf{A} es **indefinida**.

Denominamos \mathbf{g} al vector gradiente Jacobiano de la función SQD . Se define el vector gradiente \mathbf{g}^T como el transpuesto del vector gradiente ∇S , que es una matriz fila con derivadas parciales de primer orden de la función SQD :

$$\mathbf{g}^T = \left[\frac{\partial SQD}{\partial X_1} \frac{\partial SQD}{\partial X_2} \frac{\partial SQD}{\partial X_3} \dots \frac{\partial SQD}{\partial X_p} \right] \quad \text{[III.3.7]}$$

Teniendo en cuenta la definición de la función SQD dada en la ecuación [III.3.4] y derivando con respecto a cada uno de los parámetros se obtiene el vector gradiente, \mathbf{g} . La derivada de la función objetivo, SQD , con respecto a uno de los parámetros X_p será:

$$\frac{\partial SQD}{\partial X_p} = 2 \sum_1^{N_d} (f_i - y_i) \frac{\partial f_i}{\partial X_p} \quad \text{[III.3.8]}$$

Definiendo el residual individual a tiempo i como la diferencia entre el valor calculado y el experimental de la propiedad que se sigue:

$$RES = (f_i - y_i) \quad \text{[III.3.9]}$$

La ecuación [III.3.8] adoptará la siguiente forma:

$$\frac{\partial SQD}{\partial X_p} = 2 \sum_1^{N_d} \text{RES}_i \frac{\partial f_i}{\partial X_p} \quad [\text{III.3.10}]$$

El vector gradiente, \mathbf{g} , vendrá definido por:

$$\mathbf{g} = 2 \begin{bmatrix} \sum_1^{N_d} \text{RES}_i \frac{\partial f_i}{\partial X_1} \\ \sum_1^{N_d} \text{RES}_i \frac{\partial f_i}{\partial X_2} \\ \sum_1^{N_d} \text{RES}_i \frac{\partial f_i}{\partial X_3} \\ \vdots \\ \sum_1^{N_d} \text{RES}_i \frac{\partial f_i}{\partial X_p} \end{bmatrix} \quad [\text{III.3.11}]$$

La matriz Hessiana, H , es una matriz simétrica $N_p \times N_p$ con las derivadas parciales de segundo orden de la función SQD :

$$H = \begin{bmatrix} \frac{\partial^2 SQD}{\partial X_1^2} & \frac{\partial^2 SQD}{\partial X_1 \partial X_2} & \frac{\partial^2 SQD}{\partial X_1 \partial X_3} & \cdots & \frac{\partial^2 SQD}{\partial X_1 \partial X_p} \\ \frac{\partial^2 SQD}{\partial X_2 \partial X_1} & \frac{\partial^2 SQD}{\partial X_2^2} & \frac{\partial^2 SQD}{\partial X_2 \partial X_3} & \cdots & \frac{\partial^2 SQD}{\partial X_2 \partial X_p} \\ \frac{\partial^2 SQD}{\partial X_3 \partial X_1} & \frac{\partial^2 SQD}{\partial X_3 \partial X_2} & \frac{\partial^2 SQD}{\partial X_3^2} & \cdots & \frac{\partial^2 SQD}{\partial X_3 \partial X_p} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 SQD}{\partial X_p \partial X_1} & \frac{\partial^2 SQD}{\partial X_p \partial X_2} & \frac{\partial^2 SQD}{\partial X_p \partial X_3} & \cdots & \frac{\partial^2 SQD}{\partial X_p^2} \end{bmatrix} \quad [\text{III.3.12}]$$

Considerando la definición de SQD , ecuación [III.3.4] y la expresión de la primera derivada [III.3.10]:

$$\frac{\partial^2 SQD}{\partial X_s^2} = 2 \sum_1^{N_d} \left\{ \left(\frac{\partial f_i}{\partial X_s} \right) \left(\frac{\partial f_i}{\partial X_s} \right) + \text{RES}_i \frac{\partial^2 f_i}{\partial X_s^2} \right\} \quad [\text{III.3.13}]$$

$$\frac{\partial^2 SQD}{\partial X_s \partial X_v} = 2 \sum_1^{N_d} \left\{ \left(\frac{\partial f_i}{\partial X_s} \right) \left(\frac{\partial f_i}{\partial X_v} \right) + \text{RES}_i \frac{\partial^2 f_i}{\partial X_s \partial X_v} \right\} \quad [\text{III.3.14}]$$

donde RES_i viene definido por la ecuación [III.3.9], N_p es el número máximo de parámetros y el significado de los subíndices es el siguiente: $s = 1, 2, 3, \dots, N_p$; $v = 1, 2, 3, \dots, N_p$.

Se puede despreciar el segundo sumando de la expresión [III.3.14] sin cometer demasiado error y se llega a:

$$\mathbf{H} = 2 \begin{bmatrix} \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_1} \right)^2 & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_1} \frac{\partial f_i}{\partial X_2} \right) & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_1} \frac{\partial f_i}{\partial X_3} \right) & \cdots & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_1} \frac{\partial f_i}{\partial X_p} \right) \\ \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_2} \frac{\partial f_i}{\partial X_1} \right) & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_2} \right)^2 & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_2} \frac{\partial f_i}{\partial X_3} \right) & \cdots & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_2} \frac{\partial f_i}{\partial X_p} \right) \\ \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_3} \frac{\partial f_i}{\partial X_1} \right) & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_3} \frac{\partial f_i}{\partial X_2} \right) & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_3} \right)^2 & \cdots & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_3} \frac{\partial f_i}{\partial X_p} \right) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_p} \frac{\partial f_i}{\partial X_1} \right) & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_p} \frac{\partial f_i}{\partial X_2} \right) & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_p} \frac{\partial f_i}{\partial X_3} \right) & \cdots & \sum_1^{N_d} \left(\frac{\partial f_i}{\partial X_p} \right)^2 \end{bmatrix} \quad [\text{III.3.15}]$$

Los métodos de optimización de gradiente están basados en el desarrollo en serie de Taylor de la función objetivo SQD ⁽¹³²⁻¹³⁵⁾, ya que la forma explícita de dicha función es desconocida analíticamente:

$$SQD(\mathbf{X} + \Delta\mathbf{X}) \approx SQD(\mathbf{X}) + \mathbf{g}^T \Delta\mathbf{X} + \frac{1}{2} \Delta\mathbf{X}^T \mathbf{H} \Delta\mathbf{X} + \dots \quad [\text{III.3.16}]$$

Parámetros buscados: $\mathbf{X} = (X_1, \dots, X_p)$

$\Delta\mathbf{X}$ = vector de movimiento, constituido por las variaciones de los valores de los parámetros que se están optimizando en el proceso.

\mathbf{g} = vector gradiente Jacobiano de la función SQD cuyos elementos son las derivadas parciales de SQD con respecto a cada parámetro.

\mathbf{g}^T = transpuesto del vector gradiente (\mathbf{g}).

\mathbf{H} = Matriz Hessiana de la función SQD, matriz simétrica $N_p \times N_p$ con las derivadas parciales de segundo orden de la función SQD.

Si en la ecuación [III.3.16] se desprecian los términos en los que intervienen las derivadas superiores al segundo orden, tendremos la siguiente expresión:

$$SQD(\mathbf{X} + \Delta\mathbf{X}) \approx SQD(\mathbf{X}) + \mathbf{g}^T \Delta\mathbf{X} + \frac{1}{2} \Delta\mathbf{X}^T \mathbf{H} \Delta\mathbf{X} \quad [\text{III.3.17}]$$

En los métodos de optimización que desprecian el último término de la ecuación [III.3.17] se denominan métodos de primer orden, como es el caso del método de “*Steepest Descent*”. La función en este caso viene expresada por:

$$SQD(\mathbf{X} + \Delta\mathbf{X}) \approx SQD(\mathbf{X}) + \mathbf{g}^T \Delta\mathbf{X} \quad [\text{III.3.18}]$$

En los métodos de optimización que utilizan las segundas derivadas, denominados métodos de segundo orden, la función SQD viene expresada por el desarrollo en serie dado por la ecuación [III.3.17]. Algoritmos de gradiente de segundo orden son por ejemplo los algoritmos Gauss-Newton, Levenberg-Marquardt... y el algoritmo AGDC.

Dado que el objetivo que se pretende, es la minimización de la función SQD, se derivan las ecuaciones [III.3.17] y [III.3.18], según se considere un método de primer o segundo orden, con respecto a cada uno de los parámetros y se anula la derivada. En la resolución se obtiene el punto \mathbf{X}^* en el cual la función SQD tiene un valor mínimo.

En el caso de un método de primer orden ⁽¹³²⁾ (“*Steepest Descent*”) se obtendrá:

$$\mathbf{X}^* \approx \mathbf{X} - \mathbf{g}(\mathbf{X}) \quad [\text{III.3.19}]$$

y en el caso de los métodos de segundo orden:

$$\mathbf{X}^* \approx \mathbf{X} - \mathbf{H}(\mathbf{X})^{-1} \mathbf{g}(\mathbf{X}) \quad [\text{III.3.20}]$$

Mediante estas ecuaciones podemos movernos iterativamente hacia el mínimo, \mathbf{X}^* , considerando una estimación inicial $\mathbf{X}^{(0)}$ y generando valores $\mathbf{X}^{(m)}$ mediante las correspondientes expresiones dependiendo del método empleado.

En el caso del método de “*Steepest Descent*” a partir de la ecuación [III.3.19] se obtiene:

$$\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} - \mathbf{g}(\mathbf{X}^{(m)}) \quad m = \text{número de iteración } (0,1, \dots) \quad \text{[III.3.21]}$$

El vector de movimiento será:

$$\mathbf{p}^{(m)} = \Delta\mathbf{X}^{(m)} = -\mathbf{g}(\mathbf{X}^{(m)}) \quad \text{[III.3.22]}$$

En el caso de los métodos de segundo orden, a partir de la ecuación [III.3.20], se obtiene la expresión:

$$\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} - \mathbf{H}(\mathbf{X}^{(m)})^{-1} \mathbf{g}(\mathbf{X}^{(m)}) \quad m = (0,1, \dots) \quad \text{[III.3.23]}$$

El vector movimiento en este caso es:

$$\mathbf{p}^{(m)} = \Delta\mathbf{X}^{(m)} = -\mathbf{H}(\mathbf{X}^{(m)})^{-1} \mathbf{g}(\mathbf{X}^{(m)}) \quad \text{[III.3.24]}$$

En los métodos de gradiente el vector movimiento en cada iteración ($\mathbf{p}^{(m)} = \Delta\mathbf{X}^{(m)}$) debe alcanzar un punto $\mathbf{X}^{(m+1)}$ que proporcione un valor inferior de la función *SQD* y asegure el descenso controlado, evitando de este modo la divergencia. El método “*Steepest Descent*” es un método que asegura la convergencia hacia el mínimo ⁽¹³²⁾, pero su evolución hacia el mismo es excesivamente lenta ⁽¹²¹⁻¹²³⁾.

El método de Gauss-Newton ⁽¹³²⁻¹³⁵⁾, que es un método de gradiente de segundo orden que utiliza como vector de movimiento el dado por la ecuación [III.3.24], converge cuadráticamente, sin embargo, debido a las aproximaciones que se imponen y a la condición no estrictamente cuadrática de la función objetivo, presenta varios problemas que pueden conducir al fracaso del proceso. Los problemas más frecuentes son:

1. Si la matriz Hessiana, $\mathbf{H}^{(m)}$, es singular; $|\mathbf{H}^{(m)}| = 0$, no existe su inversa $(\mathbf{X}^{(m)})^{-1}$ por lo que no podremos definir el vector movimiento.
2. Si $\mathbf{H}^{(m)}$ es “*quasi*” singular o está mal condicionada, entonces no se realiza correctamente la inversión de la matriz y los resultados obtenidos son incorrectos.
3. La matriz Hessiana $\mathbf{H}^{(m)}$ no es necesariamente definida positiva, ya que ésta se calcula en un punto distinto del mínimo, luego podría ser que el proceso no sea convergente. Esto ocurre frecuentemente y con mayor probabilidad en los puntos alejados del mínimo, puesto que en ellos no es válida la aproximación cuadrática de la función.
4. Aún siendo $\mathbf{H}^{(m)}$ definida positiva y existiendo su inversa, puede que $SQD^{(m+1)} > SQD^{(m)}$ debido a que el módulo $\mathbf{p}^{(m)}$ no sea el adecuado. (Esto suele suceder en los puntos alejados del mínimo).

Existen métodos de gradiente que modifican el método de Gauss-Newton con el fin de mejorar los resultados, como por ejemplo el método de Levenberg-Mardquardt que modifica

el vector $\mathbf{p}^{(m)}$ mediante la introducción de un parámetro $\lambda^{(132-135)}$, o los métodos quasi-Newton, como el método Davidon-Fletcher-Powell, que determina una matriz inversa aproximada mediante aproximaciones sucesivas.

III.3.2.- Algoritmo AGDC

En el presente trabajo para llevar a cabo la optimización se ha utilizado el algoritmo AGDC (Algoritmo General de Descenso Controlado) ^(65-74,115-131), se trata de un algoritmo de gradiente de segundo orden en el que la dirección de búsqueda que se sigue inicialmente es la del método de Gauss-Newton ⁽¹³²⁻¹³⁵⁾, ya que éste aporta una rápida velocidad de convergencia en las proximidades de un punto crítico, pero se introducen dispositivos de control para evitar la divergencia.

Este algoritmo posee numerosas vías alternativas, mecanismos de control y las pertinentes modificaciones en caso de fallo que subsanan las limitaciones que existen en el método de Gauss-Newton. Además posee una gran versatilidad que le confiere unos amplios límites de aplicación a multitud de funciones procedentes del campo de la química, sin más que modificar ligeramente su estructura en orden a adecuarla a las características propias de las funciones y requisitos que imponen los sistemas.

A continuación enumeraremos las soluciones aportadas por el AGDC para las deficiencias que presenta el Gauss-Newton:

1. Si la matriz Hessiana $\mathbf{H}^{(m)}$ es singular, su inversa no existe y no podemos definir el vector de movimiento $\mathbf{p}^{(m)}$. En este caso seguimos la dirección del método “*Steepest Descent*” ⁽¹³⁶⁻¹³⁷⁾, luego $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$.
2. Si la matriz Hessiana no es definida positiva, la dirección de Gauss-Newton será divergente, $SQD^{(m+1)} > SQD^{(m)}$. En este caso se sustituye $\mathbf{p}^{(m)}$ por $-\mathbf{p}^{(m)}$.
3. Si la dirección $\mathbf{p}^{(m)}$ es ortogonal a $\mathbf{g}^{(m)}$, las direcciones de $\mathbf{p}^{(m)}$ y de $-\mathbf{p}^{(m)}$ no conducen a una disminución de $SQD(X)$. En este caso $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ ya que la dirección $-\mathbf{g}^{(m)}$ siempre es de descenso.

Una vez controlada la dirección del vector de movimiento, es necesario controlar la longitud del mismo mediante el parámetro $\alpha^{(m)}$, que puede determinarse a partir del **Método de Hartley** ⁽¹³⁴⁾, el cual aproxima la función que se está minimizando a un polinomio de segundo grado:

$$\alpha^{(m)} = \frac{-2(\mathbf{g}^{(m)})^T \mathbf{p}^{(m)}}{(\mathbf{p}^{(m)})^T \mathbf{H}^{(m)} \mathbf{p}^{(m)}} \quad \text{[III.3.25]}$$

El criterio de convergencia que debe cumplir $\alpha^{(m)}$, para asegurar que se produzca un decrecimiento suficiente, es el criterio de **Goldstein-Armijo** ⁽¹³⁴⁾:

$$SQD(X^{(m)}) - SQD(X^{(m)} + \alpha \mathbf{p}^{(m)}) \geq -\mu \alpha \mathbf{g}^{(m)T} \mathbf{p}^{(m)} \quad \text{[III.3.26]}$$

donde el escalar μ debe estar comprendido entre $0 < \mu \leq 0.5$

Vamos a exponer a continuación, la estructura detallada de este algoritmo. Partimos de una estimación $\mathbf{X}^{(0)}$ del minimizador \mathbf{X}^* .

1. $m = 0$.
2. Calcular $\mathbf{g}^{(m)}$ y $\mathbf{H}^{(m)}$.
3. Si $\mathbf{H}^{(m)}$ es singular entonces $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ e ir a 8.
4. Calcular $(\mathbf{H}^{(m)})^{-1}$.
5. Calcular $\mathbf{p}^{(m)}$ por: $\mathbf{p}^{(m)} = -(\mathbf{H}^{(m)})^{-1}\mathbf{g}^{(m)}$
6. Si $|(\mathbf{p}^{(m)})^T \mathbf{g}^{(m)}| < \varepsilon$, siendo ε un número arbitrario muy pequeño, entonces: $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ e ir a 8.
7. Si $\mathbf{p}^{(m)} \mathbf{g}^{(m)} > 0$ entonces $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$.
8. Calcular $\alpha^{(m)}$ mediante la ecuación [III.3.25].
9. Cálculo de $\mathbf{X}^{(m+1)}$ por: $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \alpha^{(m)}\mathbf{p}^{(m)}$
10. Si se cumple la condición de Goldstein-Armijo, ecuación [III.3.26], seguir.
11. Si se cumplen los criterios de convergencia ir a 13.
12. $m = m + 1$ e ir a 2.
13. $\mathbf{X}^* = \mathbf{X}^{(m+1)}$.

En el algoritmo **AGDC** los dispositivos de control del vector de movimiento van a resultar determinantes en el éxito del proceso de optimización, con relación a las metodologías basadas en algoritmos de gradiente de segundo orden como el de Gauss-Newton.

III.3.3.- Establecimiento y resolución del sistema de ecuaciones diferenciales de velocidad.

El algoritmo de optimización realiza la búsqueda del valor mínimo de la función SQD, que nos proporcionará el valor de los parámetros que pretendemos determinar. La función SQD no se conoce explícitamente, pero puede calcularse su valor numérico como diferencia entre los valores de las magnitudes medidas experimentalmente y las calculadas teóricamente a partir de las estimaciones iniciales de los parámetros y las que se van determinando iterativamente a lo largo del proceso de optimización. Tal y como hemos señalado anteriormente, la función SQD adopta la forma, mediante la ecuación [III.3.4]:

$$SQD = \sum_{i=1}^{N_d} (f_i(X_1, X_2, \dots, X_p, t_i) - (y_i)_E)^2$$

siendo f_i la función analítica que proporciona el valor teórico de la propiedad (concentración, absorbancia,..) e $(y_i)_E$ es el valor experimental de la propiedad obtenida a los distintos valores de tiempo. La propiedad elegida para seguir la evolución de un sistema cinético puede ser la concentración o cualquier magnitud relacionada con ella, por lo que para el cálculo de la función SQD es necesario conocer la concentración de todas las especies que intervienen en la reacción en el intervalo de tiempo considerado. Para lo cual se lleva a cabo la resolución del sistema de ecuaciones diferenciales de velocidad, que reflejan la variación de la concentración con el tiempo tal y como veremos en el siguiente apartado.

En el campo de la Química es frecuente encontrar sistemas de ecuaciones diferenciales en las cuales hay una variable independiente junto con variables dependientes, denominando al conjunto sistema de ecuaciones diferenciales ordinarias (ODE).

La solución general de una ecuación diferencial es la función que satisface la ecuación diferencial y que cumple a la vez ciertas condiciones iniciales de la función. Una ecuación diferencial puede resolverse analíticamente, pero no todas poseen solución analítica, por ello, en ocasiones es necesario recurrir a métodos numéricos aproximados.

Al realizar un estudio cinético de un determinado sistema químico, si éste es sencillo, las ecuaciones diferenciales de velocidad que proporcionan la variación de concentración con el tiempo, en general poseen solución analítica exacta. En el caso de que el sistema químico sea complejo, la solución analítica de dicho sistema de ecuaciones diferenciales es complicada, en muchos casos no existe y es necesario recurrir a métodos numéricos aproximados. En todo caso y a pesar de que los sistemas cinéticos poseen solución general explícita, nosotros vamos a utilizar la solución numérica, puesto que sin menoscabo de las garantías de exactitud que en la actualidad poseen los métodos numéricos, nos proporcionan un extenso margen de aplicabilidad, muy útil en el tratamiento de cualquier sistema o modelo cinético.

Se han descrito numerosos métodos para la resolución numérica de ecuaciones diferenciales ⁽¹³⁸⁾, entre los métodos más utilizados para la resolución de ecuaciones diferenciales destacan el método de Euler "*Predictor-Corrector*", métodos de Runge-Kutta (distintos órdenes) y como más eficiente el método de Gear ⁽¹³⁸⁾.

a) Método de Euler

Consideremos la ecuación diferencial: $\frac{dy}{dx} = f(x, y)$ cuyas condiciones iniciales son $y(x_0) = y_0$ y la función $f(x, y)$ tiene derivadas continuas en el intervalo de integración deseado. El objetivo es encontrar los valores y_1, y_2, \dots en los puntos x_1, x_2, \dots , separados por el intervalo constante h . La función $y(x)$, solución exacta de la ecuación diferencial, se desarrolla en la serie de Taylor en el punto $x = x_0$, obteniéndose:

$$y(x) = y(x_0) + (x - x_0)y'(x_0) + \frac{(x - x_0)^2 y''(x_0)}{2!} + \dots \quad \text{[III.3.27]}$$

Si las derivadas $y^{(k)}=f^{(k+1)}(x,y) = 0$ para todo $k > j$, entonces se puede calcular la solución exacta en cualquier punto desarrollando en serie hasta el término de orden j ,
$$\frac{(x-x_0)^j y^{(j)}(x_0)}{j!}$$

En el punto $x = x_1$:

$$y(x_1) = y(x_0) + hy'(x_0) + \frac{h^2 y''(x_0)}{2!} + \dots + \frac{h^j y^{(j)}(x_0)}{j!} \tag{III.3.28}$$

siendo $h=x_1-x_0$.

En la mayoría de los casos las derivadas no se anulan por lo que es necesario truncar la serie en un determinado punto. Para el punto $x = x_0$ se puede calcular $y(x_0)$, $y'(x_0)$ y las derivadas de orden superior a partir de las condiciones iniciales. Debido al truncamiento de la serie se calcula una aproximación numérica de y_1 en lugar de la solución verdadera $y(x_1)$, conocida la aproximación numérica y_1 se puede evaluar $y'_1 = f(x_1, y_1)$ y las derivadas de órdenes superiores. Operando de esta manera para el punto y_{r+1} , la solución viene dada por:

$$y_{r+1} = y_r + hy'_r + \frac{h^2 y''_r}{2!} + \dots \tag{III.3.29}$$

Existe un error de truncamiento, si la serie se trunca detrás del término $y_r^{(m)}$:

$$e_{r+1} = \frac{h^{m+1} y^{(m+1)}(\epsilon)}{(m+1)!}; \quad x_r < \epsilon < x_{r+1} \tag{III.3.30}$$

La aplicabilidad de la ecuación anterior depende de la facilidad para calcular las derivadas de segundo orden y de órdenes superiores, si y' es únicamente función de x o y , la dificultad no es grande, pero a menudo y es función de ambas variables y será necesario emplear derivadas parciales. En la mayor parte de los casos el cálculo de estas derivadas es muy complejo, por lo que se realiza el truncamiento de la serie de Taylor después del término que contiene la primera derivada, obteniéndose:

$$y_{r+1} = y_r + hy'_r = y_r + hf(x_r, y_r) \tag{III.3.31}$$

Esta ecuación se utiliza como fórmula de recurrencia para el método de Euler básico, se calcula y_1 puesto que $y(x_0)$ es conocido a través de la condición inicial $y(x_0)=y_0$, evaluándose $y'(x_0)$ a partir de $f(x_0, y_0)$ dado por la ecuación $dx/dy = f(x,y)$. Una vez que se ha calculado $y_1 = y(x_0+h)$, se calculan los valores $y(x)$ para $x_2 = x_0 + 2h, \dots$ de la misma manera. El error será:

$$e_{r+1} = \frac{h^2 y^2(\epsilon)}{2!}; \quad x_r < \epsilon < x_{r+1} \tag{III.3.32}$$

El error global de la solución es del orden de h , el error tenderá a cero si h tiende a cero.

Es un método muy sencillo pero poco exacto, el tamaño del intervalo tiene que ser muy pequeño para que la exactitud del resultado sea aceptable, lo cual implica un gran número de pasos y un tiempo de cálculo muy largo.

El método de Euler utiliza el valor de la pendiente al inicio del intervalo y'_r , el método se puede mejorar utilizando el valor medio de la pendiente al principio y al final del intervalo:

$$y_{r+1} = y_r + h \frac{(y'_r + y'_{r+1})}{2} \quad \text{[III.3.33]}$$

La derivada es función de x e y , por lo que para evaluar y'_{r+1} se necesita conocer y_{r+1} , por lo que el método de Euler modificado (Predictor-Corrector) calcula en primer lugar un valor estimado de y_{r+1} mediante la ecuación [III.3.32], a continuación se evalúa y'_{r+1} y mediante la ecuación [III.3.33] se obtiene el valor de y_{r+1} mejorado.

b) Métodos de Runge-Kutta

Estos métodos tienen las siguientes características:

- Son métodos de paso sencillo.
- Son acordes con el método de series de Taylor, desarrollado por encima del término h^j , el orden j difiere de unos métodos a otros.
- Utilizan sólo valores de la función y no requieren la evaluación de ninguna de sus derivadas.

El más utilizado es el de cuarto orden que evalúa y_{r+1} en el punto x_{r+1} mediante la ecuación:

$$y_{r+1} = y_r + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} \quad \text{[III.3.34]}$$

$$k_1 = hf(x_r, y_r) \quad \text{[III.3.35]}$$

$$k_2 = hf(x_r + \frac{h}{2}, y_r + \frac{k_1}{2}) \quad \text{[III.3.36]}$$

$$k_3 = hf(x_r + \frac{h}{2}, y_r + \frac{k_2}{2}) \quad \text{[III.3.37]}$$

$$k_4 = hf(x_r + h, y_r + k_3) \quad \text{[III.3.38]}$$

Mediante la ecuación [III.3.34] partiendo de las condiciones iniciales $y(x_0)=y_0$ se puede conocer y_1 en el punto x_1 y sucesivamente se pueden calcular y_2, y_3, \dots en los puntos x_2, x_3, \dots .

El error es del orden de h^5 frente al de Euler que es del orden de h^2 y se requieren menos evaluaciones de la función.

c) Método de Gear

El algoritmo AGDC utiliza el método de Gear, puesto que es el que proporciona resultados más satisfactorios⁽¹¹⁴⁾ y es adecuado para sistemas de reacciones químicas que tienen constantes de velocidad de muy diferente magnitud (sistemas con características “*stiff*”).

Haremos una exposición breve del algoritmo Gear⁽¹³⁸⁾ y para ello debemos considerar la ecuación diferencial:

$$\frac{dy}{dx} = f(x, y) \text{ siendo las condiciones iniciales } y(x_0) = y_0$$

El método de Gear es un método de paso múltiple, es decir, para calcular la solución y_{r+1} en el punto x_{r+1} es necesario conocer y_r, y_{r-1}, \dots en los puntos x_r, x_{r-1}, \dots . En este algoritmo se calcula en primer lugar el valor y_{m+1} en función de los p valores $y_m, y_{m-1}, \dots, y_{m-p+1}$ de los puntos anteriores:

$$y_{m+1} = \alpha_1 y_m + \alpha_2 y_{m-1} + \dots + \alpha_p y_{m-p+1} + \alpha_p h f(y_m) \quad \text{[III.3.39]}$$

El valor de y_{m+1} se mejora mediante la fórmula implícita:

$$y_{m+1} = \beta_1 y_m + \beta_2 y_{m-1} + \dots + \beta_p y_{m-p+1} + \beta_p h f(y_m) \quad \text{[III.3.40]}$$

Los valores de los coeficientes α_i y β_i pueden determinarse mediante el algoritmo Gear, éste algoritmo es considerado como “*self-starting*” debido a que él mismo genera los primeros puntos, a diferencia de la mayoría de los métodos “*predictor-corrector*” en los cuales es necesario generar los primeros puntos por otros métodos.

El orden “ p ” es variable y se modifica su valor cuando se estima necesario. El tamaño del intervalo h también es variable y se determina mediante la exactitud predeterminada.

Para resolver la ecuación [III.3.40] es necesario utilizar un método iterativo, esta ecuación para y_{m+1} se puede escribir de forma compacta como:

$$y = g + \beta f(y) \quad \text{[III.3.41]}$$

donde se omiten los subíndices $m+1$ y $p+1$. Por lo tanto se requiere el valor de y para el cual:

$$F(y) = y - \beta h f(y) - g = 0 \quad \text{[III.3.42]}$$

Este valor se puede obtener utilizando el método de Newton, mediante el cual se calcula una estimación de $y^{(s+1)}$ a partir de otra previa según:

$$y^{(s+1)} = y^{(s)} - \frac{F(y^{(s)})}{F'(y^{(s)})} \quad \text{[III.3.43]}$$

En el estudio de sistemas químicos complejos, es interesante conocer las concentraciones de todas las especies, por lo tanto la ecuación [III.3.41] para una especie individual se reemplaza por una ecuación vectorial de la forma:

$$\mathbf{y} = \mathbf{g} + \beta \mathbf{f}(\mathbf{y}) \quad \text{[III.3.44]}$$

en la que hemos utilizado notación “**bold type**” para referirnos a vectores y/o matrices.

El vector \mathbf{y} contiene las concentraciones de todas las especies en el tiempo $m+1$. La ecuación para el método de Newton en este caso:

$$\mathbf{y}^{(s+1)} = \mathbf{y}^{(s)} - \mathbf{J}^{-1}(\mathbf{y}^{(s)}) \mathbf{F}(\mathbf{y}^{(s)}) \quad \text{[III.3.45]}$$

donde $\mathbf{J}(\mathbf{y})$ es la matriz Jacobiana que contiene las derivadas $\partial \mathbf{F}(\mathbf{y}) / \partial \mathbf{y}$. Aplicando la ecuación [III.3.42] obtenemos:

$$\mathbf{y}^{(s+1)} = \mathbf{y}^{(s)} - \left[1 - h\beta \frac{\partial \mathbf{f}(\mathbf{y}^{(s)})}{\partial \mathbf{y}} \right]^{-1} [\mathbf{y}^{(s)} - h\beta \mathbf{f}(\mathbf{y}^{(s)}) - \mathbf{g}] \quad \text{[III.3.46]}$$

Para un sistema complejo la matriz:

$$1 - h\beta \frac{\partial \mathbf{f}}{\partial \mathbf{y}} \quad \text{[III.3.47]}$$

contendrá un número escaso de elementos distintos de cero, es decir, será una matriz del tipo denominado “*sparse*” que puede ahorrar mucho tiempo de computación, si se tiene en cuenta en el cálculo de la matriz inversa. La robustez del algoritmo de Gear asegura la resolución correcta de sistemas de ecuaciones diferenciales con características marcadamente “*stiff*”.

III.3.4.- Derivación Numérica

En todo método de optimización matemática de gradiente, es necesario realizar el cálculo de las derivadas de la función objetivo respecto a los parámetros que hemos de determinar. Para la aplicación del algoritmo AGDC es necesario el cálculo del vector gradiente y la matriz Hessiana, para cuya determinación es necesario el cálculo de las derivadas de la función con respecto a cada uno de los parámetros, que son las variables de un proceso de optimización matemática.

Si la función objetivo se conoce explícitamente, existen fórmulas que expresan la derivada de dicha función y por lo tanto se puede conocer analíticamente. En nuestro caso, la función objetivo no se conoce explícitamente y para determinar su derivada es necesario utilizar el cálculo numérico.

Para llevar a cabo el cálculo numérico de las derivadas se emplea el método de Diferencias Centrales ⁽¹³⁸⁾, que proporciona resultados dotados de una gran exactitud ⁽¹²¹⁾ cuando se elige un tamaño del intervalo adecuado. De acuerdo con este método, la primera derivada de una función f en un punto x_0 será:

$$f'(x_0) = \frac{f(x_0+h)-f(x_0-h)}{2h} + 0(h^2) \quad \text{[III.3.48]}$$

donde h es el tamaño del intervalo y $0(h^2)$ es el orden del error cometido.

III.3.5.- Cálculo de la matriz inversa

El cálculo del vector movimiento implica la determinación de la inversa de la matriz Hessiana. Existen numerosos métodos para el cálculo de la inversa de una matriz ⁽¹³⁹⁻¹⁴⁰⁾, de los cuales empleamos el método de eliminación de Gauss, para calcular en primer lugar una matriz inversa que llamaremos aproximada.

El método de eliminación de Gauss reduce el proceso de inversión de la matriz, mediante una serie de operaciones elementales a una forma triangular superior, calculando a continuación los elementos de la matriz inversa por recurrencia.

Expresado de forma algorítmica el proceso de eliminación es el siguiente:

Para $m = 1, 2, \dots, n - 1$:

1. Determinar el pivote a_{1m} tal que $|a_{1m}| \geq |a_{im}|$ para $i = m, \dots, n$ ($l \geq m$). Almacenar el número $l(m)$ de la fila del pivote (para cada m).
2. $a_{mj} \leftrightarrow a_{ij}$ (con $l = l(m)$), para $j = m, \dots, n$. Para $i = m + 1, \dots, n$.
3. $k_{im} = -a_{im} / a_{mm}$
4. $a_{ij} := a_{ij} + k_{im}a_{mj}$ (para $j = m + 1, \dots, n$).
5. $b_m \leftrightarrow b_1$ (con $l = l(m)$)
6. $b_i := b_i + k_{im}b_m$

$$7. x_n = b_n / a_{nn}$$

Para $i = n, \dots, 1$:

$$x_i = \frac{b_i - \sum_j a_{ij}x_j}{a_{ii}} \quad j = i + 1, \dots, n$$

Para mejorar la exactitud de la matriz inversa, es posible utilizar un método iterativo de aproximaciones sucesivas ⁽¹⁴⁰⁾. Consideremos una medida del error de la inversa de una matriz de la forma:

$$F_0 = I - A D_0 \quad \text{[III.3.49]}$$

donde:

A = matriz que se desea invertir D_0 = matriz inversa

I = matriz unidad F_0 = error

Si F_0 es igual a cero, entonces D_0 es la inversa exacta de la matriz A , pero si no lo es, D_0 no es la inversa exacta y podemos formular aproximaciones sucesivas a partir de la siguiente expresión:

$$D_m = D_{m-1} + D_{m-1} F_{m-1} \quad (m = 1, 2, 3 \dots) \quad \text{[III.3.50]}$$

Los errores correspondientes son:

$$F_m = I - A D_m \quad \text{[III.3.51]}$$

Empleando este método iterativo, que normalmente converge rápidamente, se puede obtener una matriz inversa mejorada. El proceso iterativo se termina cuando se ha conseguido la desigualdad:

$$\|D_m - D_{m-1}\| \leq \delta \quad \text{[III.3.52]}$$

La garantía de éxito en el proceso de inversión nos la proporciona el hecho de que el producto de la matriz inversa “mejorada” por la inicial, es igual a la matriz unidad en la mayoría de los casos. Este proceso de mejora de la matriz inversa asegura resultados satisfactorios incluso en los casos más desfavorables, es decir, en aquellos cuyas matrices están formadas por elementos de muy diferente orden de magnitud.

III.4.- Aspectos Teóricos en Cinética Química

A lo largo del apartado anterior hemos visto la capacidad que tiene la metodología AH para el análisis y tratamiento de una gran cantidad de información. Teniendo en cuenta que el objetivo del trabajo que recoge la presente Memoria es la determinación de parámetros cinéticos (en este primer caso constantes de velocidad) y de parámetros termodinámicos de activación (PTA) de distintos mecanismos de reacción mediante la aplicación del algoritmo AH, es necesario considerar brevemente una serie de cuestiones cinéticas de índole teórica.

En la actualidad es habitual obtener una gran cantidad de información en un intervalo de tiempo suficientemente rápido cuando se monitoriza un sistema cinético. Los métodos para la determinación de parámetros cinéticos se clasifican en dos grandes grupos: *Hard Modelling* ⁽¹⁴¹⁻¹⁴³⁾ y *Soft Modelling* ⁽¹⁴⁴⁻¹⁴⁵⁾. Las metodologías denominadas *Hard Modelling* presuponen un modelo cinético, y mediante métodos de regresión no lineal permiten determinar parámetros cinéticos, analíticos y/o termodinámicos. La aplicación de estos métodos implica conocer “*a priori*” las condiciones iniciales para cada una de las especies involucradas en un mecanismo de reacción, conduciendo a buenos resultados cuando el modelo supuesto es el correcto. Se utilizan técnicas de resolución *univariable* y *multivariable* de curvas. Por supuesto, las metodologías del tipo *Soft Modelling* son capaces de modelar de manera cualitativa y/o semicuantitativa un sistema de reacción sin necesidad de presuponer “*a priori*” un modelo cinético. Asimismo, se ha utilizado una metodología híbrida que consiste en combinar los dos procedimientos, esto es, utilizar los modelos cinéticos junto con las metodologías de resolución de curvas, para aprovechar las ventajas de ambos y de esta manera ajustar los perfiles de concentración al modelo cinético por medio de algoritmos de Regresión no lineal.

En la mayoría de las metodologías cinéticas, se estudia la variación de las concentraciones de los reactivos y/o productos con respecto al tiempo. La reacción se lleva a cabo habitualmente en condiciones isotérmicas y las constantes de velocidad se obtienen a partir de la resolución de las ecuaciones diferenciales de velocidad. En el caso de que el objetivo sea la determinación de Parámetros Termodinámicos de Activación (PTA), es necesario estudiar la influencia de la temperatura en el proceso cinético.

El estudio de la influencia de la temperatura se puede llevar a cabo mediante dos métodos distintos:

a) Método Clásico (Isotérmico), consistente en realizar una serie de experiencias cinéticas (bastante limitada en número) en la que la reacción evoluciona a temperatura constante (condiciones isotérmicas) y cada experiencia cinética se desarrolla a una temperatura distinta a las demás. Teniendo en cuenta el pequeño margen experimental de variación de temperatura que puede conseguirse en el laboratorio, no es posible obtener un colectivo muy numeroso de datos.

b) Método no isotérmico, consiste en imponer una variación continua de la temperatura en el transcurso de la cinética de la reacción lo cual supone introducir una variable más (temperatura). La ventaja es que el colectivo de datos cinéticos a evaluar puede ser muy numeroso si tenemos en cuenta la adquisición “*on line*” de los datos cinéticos. La función temperatura vs tiempo (T vs t) debe corresponder a una función continua, monótona creciente y debe responder a un perfil de variación de una función adecuada que pueda conseguirse en el laboratorio. Ahora bien la inclusión de una nueva variable cinética es causa de serias dificultades de tratamiento en dos aspectos: a) Teórico, surgen fuertes complicaciones matemáticas de tratamiento del sistema de ecuaciones diferenciales, de integración, etc,...

b) Experimental, es necesario reproducir fielmente el perfil de variación temporal de temperatura de acuerdo a la naturaleza de la función matemática elegida.

III.4.1.-Aspectos Teóricos en Cinética Química Isotérmica

Uno de los objetivos del trabajo que recoge la presente memoria es la aplicación de un Algoritmo Híbrido (AH) constituido por dos metodologías diferentes ANN y AGDC, para la determinación de parámetros cinéticos como son las constantes individuales de velocidad de diferentes mecanismos de reacción. La determinación de las constantes de velocidad de un mecanismo de reacción se lleva a cabo mediante diferentes métodos a partir del tratamiento de los datos experimentales adquiridos a temperatura constante (Cinética Isotérmica).

Consideramos un sistema químico en el que existen n_s especies químicas y en el que pueden estar involucradas n_r reacciones químicas. De acuerdo con las normas dictadas por IUPAC ⁽¹⁴⁶⁻¹⁴⁷⁾, el sistema de reacción se puede representar:

$$\begin{aligned}
 0 &= \nu_{1,1}B_1 + \nu_{2,1}B_2 + \nu_{3,1}B_3 + \dots + \nu_{n_s,1}B_{n_s} \\
 0 &= \nu_{1,2}B_1 + \nu_{2,2}B_2 + \nu_{3,2}B_3 + \dots + \nu_{n_s,2}B_{n_s} \\
 0 &= \nu_{1,3}B_1 + \nu_{2,3}B_2 + \nu_{3,3}B_3 + \dots + \nu_{n_s,3}B_{n_s} \\
 &\dots\dots\dots \\
 0 &= \nu_{1,n_r}B_1 + \nu_{2,n_r}B_2 + \nu_{3,n_r}B_3 + \dots + \nu_{n_s,n_r}B_{n_s}
 \end{aligned}$$

De manera general, podemos expresarlo para cada una de las r reacciones químicas:

$$0 = \sum_{j=1}^{n_s} \nu_{j,r} B_j \quad \text{[III.4.1]}$$

donde,

B_j = especies químicas que intervienen en el mecanismo de reacción.

$r = (1, \dots, n_r)$ número de reacciones elementales del mecanismo.

$j = (1, \dots, n_s)$ número de especies involucradas en el mecanismo.

k_r = constante de velocidad de la reacción r .

$\nu_{j,r}$ = coeficiente estequiométrico de la especie B_j en la reacción r .

$\nu_{j,r} < 0$ para las especies que actúan como reactivos en la reacción r .

$\nu_{j,r} > 0$ para las especies que actúan como productos en la reacción r .

Cada una de las r reacciones elementales o concertadas que componen el mecanismo origina, en un intervalo de tiempo, un incremento diferencial en las concentraciones de las especies B_j igual a $d[B_j]$:

$$\begin{aligned}
 r=1 \quad & \frac{d[B_j]}{dt} = k_1 v_{j,1} \prod_{l=1}^{n_s} [B_l]^{|z_{l,1}|} \\
 r=2 \quad & \frac{d[B_j]}{dt} = k_2 v_{j,2} \prod_{l=1}^{n_s} [B_l]^{|z_{l,2}|} \\
 & \dots\dots\dots \\
 r = n_r \quad & \frac{d[B_j]}{dt} = k_{n_r} v_{j,n_r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,n_r}|} \quad \text{[III.4.2]}
 \end{aligned}$$

donde B_l se refiere a las especies que desempeñan el papel de reactivo en la reacción r ($v_{l,r} < 0$). El incremento diferencial total en la concentración de B_j será la suma extendida a todas las reacciones (n_r):

$$\frac{d[B_j]}{dt} = \sum_{r=1}^{n_r} k_r v_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \quad \text{[III.4.3]}$$

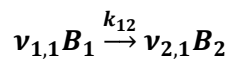
Como es conocido, para reacciones elementales o concertadas, se verifica que el coeficiente estequiométrico de la especie coincide en valor absoluto con el orden individual cinético. Esto es: $|v_{l,r}| = |z_{l,r}|$.

La resolución del sistema de ecuaciones diferenciales de velocidad proporciona la concentración de cada especie en el intervalo de tiempo considerado, ($[B_j]_{t_i}$).

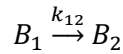
A lo largo del trabajo que recoge la presente Memoria se ha estudiado la determinación de constantes de individuales de velocidad para diferentes mecanismos de reacción de interés en Química Física. Asimismo, se ha analizado la posibilidad de discriminación entre diferentes mecanismos de reacción. Es decir, hemos evaluado la capacidad del algoritmo AH para asignar a un conjunto de datos cinéticos, el mecanismo correcto responsable de la reacción química de la cual proceden, pudiendo discriminar o distinguir entre todos los que son posibles. En este contexto, los mecanismos analizados son los siguientes:

- 1- $B_1 \xrightarrow{k_{12}} B_2$
- 2- $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$
- 3- $B_1 \xrightleftharpoons[k_{21}]{k_{12}} B_2 \xrightarrow{k_{23}} B_3$
- 4- $B_1 \xrightarrow{k_{12}} B_2 \xrightleftharpoons[k_{32}]{k_{23}} B_3$
- 5- $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3 \xrightarrow{k_{31}} B_1$

Mecanismo I



Si consideramos $v_{1,1} = -1$, $v_{2,1} = 1$, el mecanismo se puede escribir como:



donde hemos asignado un doble subíndice a las constantes cinéticas (jj') para denotar la reacción a la que corresponde, siguiendo precisamente este orden $j = \text{índice del reactivo y } j' \text{ del producto}$ a los que corresponde la reacción $B_j \rightarrow B_{j'}$.

El sistema de ecuaciones diferenciales de velocidad nos proporciona la variación de la concentración de las dos especies que intervienen en el mismo, expresado en forma matricial ⁽¹⁴⁸⁾, puede escribirse:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \end{bmatrix} = \begin{bmatrix} -k_{12} & 0 \\ k_{12} & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \end{bmatrix} \quad \text{[III.4.4]}$$

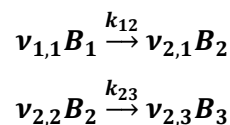
En el caso del modelo cinético sencillo que implica el paso o transformación de un compuesto B_1 para dar un compuesto B_2 mediante una reacción sencilla de orden 1, las ecuaciones diferenciales que dan la variación de la concentración de B_1 y B_2 con el tiempo son las siguientes:

$$\begin{aligned} \frac{d[B_1]}{dt} &= -k_{12}[B_1] \\ \frac{d[B_2]}{dt} &= k_{12}[B_1] \end{aligned} \quad \text{[III.4.5]}$$

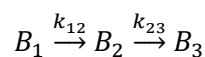
Si $[B_1]_0 \neq 0$ y $[B_2]_0 = 0$, la integración del sistema de ecuaciones diferenciales conduce a las siguientes expresiones para la concentración de B_1 y B_2 :

$$\begin{aligned} \ln \frac{[B_1]}{[B_1]_0} &= -k_{12}t \\ [B_1] &= [B_1]_0 e^{-k_{12}t} \\ [B_2] &= [B_1]_0 - [B_1] = [B_1]_0 - [B_1]_0 e^{-k_{12}t} = [B_1]_0 (1 - e^{-k_{12}t}) \end{aligned} \quad \text{[III.4.6]}$$

Mecanismo II



Si consideramos $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,2} = -1, \nu_{2,3} = 1$, podemos representar este mecanismo de la siguiente manera:



De acuerdo con el mecanismo, el sistema de ecuaciones diferenciales de velocidad que nos proporciona la variación de la concentración de las tres especies que intervienen en el mismo, expresado en forma matricial ⁽¹⁴⁸⁾, es:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12} & 0 & 0 \\ k_{12} & -k_{23} & 0 \\ 0 & k_{23} & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \quad \text{[III.4.7]}$$

Este modelo consiste en un sistema de reacciones irreversibles consecutivas de orden uno, donde la transformación del reactivo B_1 en el producto B_3 tiene lugar a través de la formación de la especie intermedia B_2 . El sistema de ecuaciones diferenciales de velocidad que ofrecen la variación de la concentración de las diferentes especies con el tiempo son las siguientes:

$$\begin{aligned} \frac{d[B_1]}{dt} &= -k_{12}[B_1] \\ \frac{d[B_2]}{dt} &= k_{12}[B_1] - k_{23}[B_2] \\ \frac{d[B_3]}{dt} &= k_{23}[B_2] \end{aligned}$$

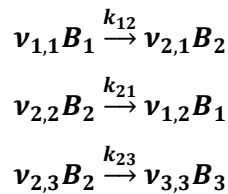
[III.4.8]

Si $[B_1]_0 \neq 0$ y $[B_2]_0 = [B_3]_0 = 0$, la integración del sistema de ecuaciones diferenciales conduce a las siguientes expresiones para la concentración de B_1 , B_2 y B_3 :

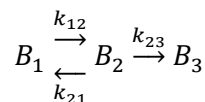
$$\begin{aligned} [B_1] &= [B_1]_0 e^{-k_{12}t} \\ [B_2] &= \frac{[B_1]_0 k_{12}}{k_{23} - k_{12}} (e^{-k_{12}t} - e^{-k_{23}t}) \\ [B_3] &= [B_1]_0 - [B_1] - [B_2] = [B_1]_0 \left(1 - \frac{k_{23}}{k_{23} - k_{12}} e^{-k_{12}t} + \frac{k_{12}}{k_{23} - k_{12}} e^{-k_{23}t} \right) \end{aligned} \quad \text{[III.4.9]}$$

La concentración de B_1 decrece exponencialmente con el tiempo desde $[B_1]_0$, la concentración del producto B_3 aumenta desde cero hasta $[B_1]_0$. La concentración de la especie intermedia B_2 aumenta desde $[B_2]_0 = 0$, alcanza un máximo y, a continuación, decrece hasta cero. La concentración de B_3 posee un punto de inflexión con el tiempo para el que la concentración de B_2 es máxima.

Mecanismo III



Si consideramos $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,2} = -1, \nu_{1,2} = 1, \nu_{2,3} = -1, \nu_{3,3} = 1$, podemos representar este mecanismo de la siguiente manera:



El sistema de ecuaciones diferenciales de velocidad, expresado de forma matricial ⁽¹⁴⁸⁾:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12} & k_{21} & 0 \\ k_{12} & -k_{21} & -k_{23} \\ 0 & k_{23} & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \quad \text{[III.4.10]}$$

En el caso del modelo de reacciones consecutivas de primer orden que consta de una etapa reversible, el sistema de ecuaciones diferenciales de velocidad es el siguiente:

$$\begin{aligned} \frac{d[B_1]}{dt} &= -k_{12}[B_1] + k_{21}[B_2] \\ \frac{d[B_2]}{dt} &= k_{12}[B_1] - k_{21}[B_2] - k_{23}[B_2] \\ \frac{d[B_3]}{dt} &= k_{23}[B_2] \end{aligned}$$

[III.4.11]

La resolución de este sistema de ecuaciones diferenciales lleva a las siguientes soluciones ⁽¹⁴⁹⁻¹⁵⁰⁾:

$$\alpha = \frac{[B_1]}{[B_1]_0} \quad \beta = \frac{[B_2]}{[B_1]_0} \quad \gamma = \frac{[B_3]}{[B_1]_0} \quad \tau = k_{12}t \quad K_1 = \frac{k_{21}}{k_{12}} \quad K_2 = \frac{k_{23}}{k_{12}}$$

$$\lambda_2 = \frac{1 + K_1 + K_2 + ((1 + K_1 + K_2)^2 - 4K_2)^{1/2}}{2}$$

$$\lambda_3 = \frac{1 + K_1 + K_2 - ((1 + K_1 + K_2)^2 - 4K_2)^{1/2}}{2}$$

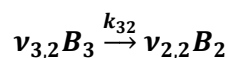
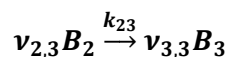
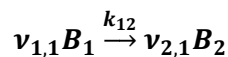
$$\alpha = \frac{[B_1]}{[B_1]_0} = \frac{\lambda_2 - K_2}{\lambda_2(\lambda_2 - \lambda_3)} e^{-\lambda_2\tau} + \frac{K_2 - \lambda_3}{\lambda_3(\lambda_2 - \lambda_3)} e^{-\lambda_3\tau}$$

$$\beta = \frac{[B_2]}{[B_1]_0} = \frac{1}{\lambda_2 - \lambda_3} (e^{-\lambda_3\tau} - e^{-\lambda_2\tau})$$

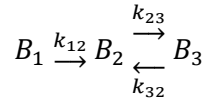
$$[B_1]_0 + [B_2]_0 + [B_3]_0 = [B_1] + [B_2] + [B_3]$$

$$[B_3] = [B_1]_0 + [B_2]_0 + [B_3]_0 - [B_1] - [B_2] \quad \text{[III.4.12]}$$

Mecanismo IV



Si consideramos $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,3} = -1, \nu_{3,3} = 1, \nu_{3,2} = -1, \nu_{2,2} = 1$, podemos representar este mecanismo de la siguiente manera:



El sistema de ecuaciones diferenciales de velocidad, expresado de forma matricial ⁽¹⁴⁸⁾:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12} & 0 & 0 \\ k_{12} & -k_{23} & k_{32} \\ 0 & k_{23} & -k_{32} \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \quad \text{[III.4.13]}$$

En este caso de reacciones consecutivas de primer orden con una etapa reversible el sistema de ecuaciones diferenciales de velocidad es el siguiente:

$$\begin{aligned} \frac{d[B_1]}{dt} &= -k_{12}[B_1] \\ \frac{d[B_2]}{dt} &= k_{12}[B_1] - k_{23}[B_2] + k_{32}[B_3] \\ \frac{d[B_3]}{dt} &= k_{23}[B_2] - k_{32}[B_3] \end{aligned}$$

[III.4.14]

Considerando que inicialmente solo está presente el reactivo B_1 , ($[B_1]_0 \neq 0$ y $[B_2]_0 = [B_3]_0 = 0$), la integración de las ecuaciones diferenciales proporciona las siguientes soluciones del sistema de ecuaciones diferenciales ⁽¹⁴⁹⁻¹⁵⁰⁾:

$$\alpha_1 = [B_1]_\infty - [B_1]_0$$

$$\alpha_2 = [B_2]_\infty - [B_2]_0$$

$$\alpha_3 = [B_3]_\infty - [B_3]_0$$

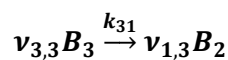
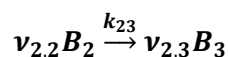
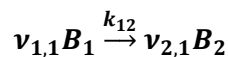
$$[B_1] = [B_1]_0 e^{-k_{12}t}$$

$$[B_2] = [B_2]_0 + \alpha_2 + \frac{1}{k_{23} + k_{32} - k_{12}} ([B_1]_0 (k_{12} - k_{32}) e^{-k_{12}t} - (\alpha_2 k_{23} - \alpha_3 (k_{32} - k_{12})) e^{-(k_{23} + k_{32})t})$$

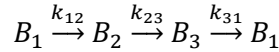
$$[B_1]_0 + [B_2]_0 + [B_3]_0 = [B_1] + [B_2] + [B_3]$$

$$[B_3] = [B_1]_0 + [B_2]_0 + [B_3]_0 - [B_1] - [B_2] \quad \text{[III.4.15]}$$

Mecanismo V



Si consideramos $\nu_{1,1} = -1, \nu_{2,1} = 1, \nu_{2,2} = -1, \nu_{2,3} = 1, \nu_{3,3} = -1, \nu_{1,3} = 1$, podemos representar este mecanismo de la siguiente manera:



El sistema de ecuaciones diferenciales de velocidad, expresado de forma matricial ⁽¹⁴⁸⁾:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12} & 0 & k_{31} \\ k_{12} & -k_{23} & 0 \\ 0 & k_{23} & -k_{31} \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \quad \text{[III.4.16]}$$

En este caso en el que las reacciones siguen un ciclo de primer orden, el sistema de ecuaciones diferenciales de velocidad es el siguiente:

$$\frac{d[B_1]}{dt} = -k_{12}[B_1] + k_{31}[B_3]$$

$$\frac{d[B_2]}{dt} = k_{12}[B_1] - k_{23}[B_2]$$

$$\frac{d[B_3]}{dt} = k_{23}[B_2] - k_{31}[B_3]$$

[III.4.17]

El modelo del Mecanismo V, no tiene solución exacta por lo que no es posible llevar a cabo la integración del sistema de ecuaciones diferenciales para así obtener las expresiones de cada una de las concentraciones. Este problema ocurre con frecuencia en mecanismos de reacción que incluyen varias etapas y especies que intervienen en diferentes etapas. Esto hace imposible integrar analíticamente las ecuaciones cinéticas simultáneas de un mecanismo. En este caso se utilizan métodos numéricos aproximados tales como los métodos de Euler, Euler "Predictor-Corrector", Runge-Kutta (distintos órdenes), Adams, Gear, etc,...

Todas las ecuaciones anteriores permiten calcular la concentración de cada una de las especies que intervienen en el sistema en un intervalo de tiempo considerado. A partir de las concentraciones podremos determinar la absorbancia de cada una de las especies a la longitud de onda de trabajo (A_{j,t_i}^λ) y la absorbancia total de la mezcla reaccionante (A_{T,t_i}^λ). De acuerdo a la ley de Lambert-Beer-Bouguer, la absorbancia de cualquier especie j al tiempo t_i y a la longitud de onda λ , medida con un paso óptico de 1 cm, viene expresada por:

$$A_{j,t_i}^\lambda = \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad \text{[III.4.18]}$$

A_{j,t_i}^λ = Absorbancia de la especie química j al tiempo t_i y a la longitud de onda λ ;

ε_j^λ = Coeficiente de absorción molar de la especie j a la longitud de onda λ ;

$[B_j]_{t_i}$ = Concentración molar de la especie B_j en el tiempo t_i .

La absorbancia total de la mezcla reaccionante al tiempo t_i y a la longitud de onda λ (A_{T,t_i}^λ), será la suma de las absorbancias individuales de las especies (A_{j,t_i}^λ) que intervienen en la reacción y que presentan absorción a la longitud de onda de trabajo, es decir:

$$A_{T,t_i}^\lambda = \sum_{j=1}^{n_s} A_{j,t_i}^\lambda = \sum_{j=1}^{n_s} \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad \text{[III.4.19]}$$

III.4.2.-Aspectos Teóricos en Cinética Química No Isotérmica

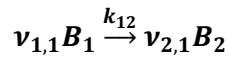
Una parte del objetivo de este trabajo es el estudio de la aplicación de la metodología basada en AH (ANN-AGDC) para la determinación de los parámetros de Arrhenius, (energía de activación, E y factor preexponencial, A) y también de los parámetros de Eyring, (variación o incremento de entalpía, ΔH^\ddagger , y variación o incremento de entropía, ΔS^\ddagger) a partir del tratamiento de datos procedentes de cinéticas no isotérmicas.

La determinación de Parámetros Termodinámicos de Activación (PTA) mediante el tratamiento de datos cinéticos no isotérmicos aplicando el AH, lo hemos llevado a cabo para los siguientes mecanismos de reacción:

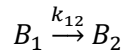
1. $B_1 \xrightarrow{k_{12}} B_2$
2. $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$

El método no isotérmico consiste en imponer una variación continua de la temperatura en el transcurso de la cinética de la reacción que supone introducir una variable más (temperatura) que permite aumentar enormemente el colectivo de datos cinéticos a evaluar. A continuación se exponen una serie de consideraciones de índole teórico sobre las cinéticas no isotérmicas en los mecanismos de reacción estudiados.

Mecanismo I



Si consideramos $\nu_{1,1} = -1$, $\nu_{2,1} = 1$, el mecanismo se puede escribir como:



El sistema de ecuaciones diferenciales de velocidad de las dos especies que intervienen en la reacción, expresado en forma matricial, es el siguiente:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \end{bmatrix} = \begin{bmatrix} -k_{12}(T) & 0 \\ k_{12}(T) & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \end{bmatrix} \quad \text{[III.4.20]}$$

En el Apartado III.4.1 se ha expuesto la ecuación [III.4.3] que proporciona el incremento total en la concentración de B_j :

$$\frac{d[B_j]}{dt} = \sum_{r=1}^{n_r} k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{\nu_{l,r}}$$

En el caso de un proceso no isotérmico, las constantes de velocidad de cada una de las etapas que constituyen un mecanismo ($k_r(T)$) son funciones de la temperatura y consecuentemente del tiempo. Si aplicamos la ecuación genérica [III.4.3] al sistema cinético sencillo correspondiente a una reacción de orden 1 (Mecanismo I) ($\nu_{1,1} = -1$), el sistema de ecuaciones diferenciales queda reducido a:

$$\frac{d[B_1]}{dt} = \nu_{1,1} k_{12}(T) [B_1] = -k_{12}(T) [B_1] \quad \text{[III.4.21]}$$

Es conveniente utilizar la variable denominada Grado de Avance de una reacción, ξ' (unidades de concentración molar) y expresar la ecuación diferencial en función de ξ' :

$$\frac{d\xi'}{dt} = k_{12}(T)([B_1]_0 - |v_{1,1}|\xi') \quad \text{[III.4.22]}$$

Separando variables e integrando posteriormente el primer miembro se obtiene la siguiente ecuación:

$$-\frac{1}{|v_{1,1}|} \ln\left(\frac{[B_1]_0 - |v_{1,1}|\xi'}{[B_1]_0}\right) = \int_0^t k_{12}(T) dt \quad \text{[III.4.23]}$$

Como ya hemos comentado anteriormente, uno de los objetivos del Trabajo que recoge la presente Memoria es la determinación de los parámetros de Arrhenius y Eyring para cada uno de los sistemas estudiados, por lo que a continuación explicaremos en detalle cada una de las ecuaciones que incluyen estos parámetros.

III.4.2.1.- Determinación de A y E. Ecuación de Arrhenius

De acuerdo con la ecuación de Arrhenius para muchas reacciones químicas la variación de k con T se ajusta a la ecuación empírica:

$$k(T) = Ae^{-\frac{E}{RT}} \quad \text{[III.4.24]}$$

siendo $A = \text{Factor preexponencial}$, $E = \text{Energía de activación}$.

En este caso resulta conveniente utilizar como variable de trabajo la fracción remanente del reactivo B_j , que vendrá dada por:

$$\alpha_1 = \frac{[B_1]}{[B_1]_0} \quad \text{[III.4.25]}$$

Sustituyendo $k(T)$ y α_1 en la ecuación [III.4.23] se obtiene la ecuación [III.4.26], a partir de la cual se puede determinar Energía de activación, E y el Factor preexponencial, A .

$$-\frac{1}{|v_{1,1}|} \ln \alpha_1 = \int_0^t A_{12} e^{-\frac{E_{12}}{RT}} dt \quad \text{[III.4.26]}$$

En condiciones no isotérmicas la temperatura (T) varía con el tiempo (t), por lo que el segundo miembro de la ecuación [III.4.26] no puede ser integrado puesto que hay dos variables dependientes ($T = f(t)$). El primer miembro puede ser integrado teniendo en cuenta el orden de la reacción; en este caso se considera una reacción de orden 1.

La función temperatura vs tiempo (T vs t) debe corresponder a una función continua, monótona creciente y respondiendo a un perfil de variación siguiendo una función adecuada de la dependencia temporal de la temperatura, ya que es necesario monitorizarla en el laboratorio.

Ahora bien la inclusión de una nueva variable cinética es causa de serias dificultades de tratamiento en dos aspectos: a) teórico: surgen serias complicaciones matemáticas de resolución exacta del sistema de ecuaciones diferenciales, de integración, etc,...b) experimental, es necesario reproducir fielmente el perfil de variación temporal de temperatura de acuerdo a la naturaleza de la función matemática elegida.

Hay múltiples posibilidades de elección de la variación continua monótonamente creciente de la temperatura. Puede ser de tipo lineal o no lineal de acuerdo a funciones genéricas que pueden ser ajustadas a expresiones polinómicas.

Si la variación de la temperatura con el tiempo es lineal ($T = T_0 + \phi t$) o polinomial ($T = T_0 + a_1 t + a_2 t^2 + \dots$) el segundo miembro de la ecuación [III.4.26] no tiene función primitiva y en consecuencia la integración no tendrá solución exacta. Si la variación de la temperatura con el tiempo es de tipo hiperbólico, la expresión resultante del segundo miembro de la ecuación [III.4.26] sí posee solución matemática exacta. Por lo tanto, es crucial establecer la identidad de la función porque el método matemático para la resolución de la ecuación [III.4.26] va a ser diferente.

La función elegida debe ser monótonamente creciente, ya que, es el camino para minimizar la gran diferencia existente en la velocidad de la reacción entre el principio y el final de la reacción. La velocidad de calentamiento debe ser la adecuada para el intervalo de tiempo estudiado y su perfil debe ser reproducible en el laboratorio. Hemos considerado varias funciones $T = f(t)$ para expresar la variación de la temperatura con el tiempo: hiperbólica, lineal y polinomial. Dependiendo del tipo de función seleccionada para expresar la variación de la temperatura, la resolución de la ecuación [III.4.26] se lleva a cabo de dos maneras diferentes: exacta y numérica.

III.4.2.1.1.- Resolución exacta

Cuando se utiliza una función $T = f(t)$ que proporciona una variación hiperbólica de la temperatura con el tiempo. La función inversa de la temperatura corresponde a la rama hiperbólica:

$$\frac{1}{T} = \frac{1}{T_0} - mt \quad \text{[III.4.27]}$$

Sustituyendo en la ecuación [III.4.26] la expresión anterior e integrando la ecuación resultante se obtiene la solución exacta, ya que, la expresión tiene función primitiva.

$$-\frac{1}{|v_{1,1}|} \ln \alpha_1 = \int_0^t A_{12} e^{\frac{-E_{12}}{R} \frac{1}{T_0} - mt} dt \quad \text{[III.4.28]}$$

Resolviendo esta integral se obtiene como solución matemática exacta la siguiente expresión:

$$\ln \alpha_1 = -\frac{|v_{1,1}| A_{12} R}{m E_{12}} e^{\frac{-E_{12}}{R} \frac{1}{T_0}} (e^{\frac{E_{12}}{R} t - 1}) \quad \text{[III.4.29]}$$

Representando de manera explícita la ecuación quedará como:

$$\alpha_1 = e^{-\frac{|v_{1,1}| A_{12} R}{m E_{12}} e^{\frac{-E_{12}}{R} \frac{1}{T_0}} (e^{\frac{m E_{12}}{R} t - 1})} \quad \text{[III.4.30]}$$

Es necesario tener en cuenta una serie de consideraciones:

- El objetivo es calcular A y E simultáneamente. Para ello hay que hacer un tratamiento de esa función con los datos experimentales.
- Hay que garantizar que esos puntos experimentales (T/t) obedecen a la función hiperbólica y además, debe ser posible llevar a cabo este experimento en el laboratorio.
- La función explícita de α dependiente del tiempo, no es una función simple. Por lo tanto, es necesario la aplicación de un método suficientemente robusto para la determinación de los parámetros termodinámicos E y A .
- Existe una gran diferencia entre los órdenes de magnitud de los dos parámetros, lo cual dificulta enormemente el tratamiento.

III.4.2.1.2.- Resolución numérica

a) Si la variación de la temperatura con el tiempo es lineal, la función $T = f(t)$ es del tipo:

$$T = T_0 + \phi t \quad \text{[III.4.31]}$$

Sustituyendo la ecuación [III.4.31] en el segundo miembro de la ecuación [III.4.26] se obtiene una nueva expresión:

$$-\frac{1}{|v_{1,1}|} \ln \alpha_1 = \int_0^t A_{12} e^{-E_{12}/R(T_0+\phi t)} dt \quad \text{[III.4.32]}$$

b) En el caso de que la variación sea polinomial de grado n , en este caso la función $T = f(t)$ es del tipo:

$$T = \sum_{j=0}^{j=n} a_j t^j \quad \text{[III.4.33]}$$

Sustituyendo esta expresión en el segundo miembro de la ecuación [III.4.26] se obtiene una nueva expresión:

$$-\frac{1}{|v_{1,1}|} \ln \alpha_1 = \int_0^t A_{12} e^{-E_{12}/R(\sum_{j=0}^{j=n} a_j t^j)} dt \quad \text{[III.4.34]}$$

Las ecuaciones [III.4.32] y [III.4.34] no tienen una solución matemática exacta, por lo que la integración no se puede llevar a cabo. Incluso para un polinomio de grado 1, es decir, una recta, esta ecuación tampoco tendría una solución exacta.

Al no existir una solución exacta para esas ecuaciones el tratamiento o método para la resolución puede ser de dos tipos:

b1) Integración numérica, los métodos numéricos de integración sustituyen la función a integrar por una aproximación y toman como valor de la integral de la función el valor de la integral de su aproximada. Existen numerosos procedimientos para integrar numéricamente una función como por ejemplo la Regla del Trapecio en la que aproxima la función a un polinomio

lineal o la Regla de Simpson que aproxima la función a un polinomio cuadrático (fórmula de *cuadratura*).

En estos métodos de manera general dada una función $f(x)$ integrable en $[a; b]$ y una aproximación a la integral dada por:

$$I(f) = \int_a^b f(x) dx \approx \sum_{i=0}^n w_i f(x_i) \quad \text{[III.4.35]}$$

donde $a \leq x_0 < x_1 \dots < x_n \leq b$

En estos métodos se dice que:

$$Q(f) = \sum_{i=0}^n w_i f(x_i) \quad \text{[III.4.36]}$$

es una fórmula de *cuadratura* de $n+1$ puntos que aproxima la integral definida. A los números w_i se les llama pesos de dicha fórmula.

Se define el error de truncamiento para la fórmula de cuadratura como:

$$E(f) = I(f) - Q(f) \quad \text{[III.4.37]}$$

y se dice que el grado (o grado de precisión) de una regla de cuadratura es K cuando consigue el resultado exacto de la integral para todos los polinomios de grado menor o igual que K , pero no así para grados mayores.

En general las fórmulas de integración numérica (como las del trapecio y de Simpson) que se obtienen por medio de la interpolación polinómica, reciben el nombre de Fórmulas de Newton–Cotes. Estas reglas garantizan la integración exacta de polinomios hasta un determinado grado dependiendo del error que lleva asociado cada una de ellas.

Existen otros métodos de integración conocidos como cuadraturas Gaussianas que integran todos los polinomios hasta un determinado grado, el más alto posible para el número de sumandos elegidos. Elegir un mayor número de sumandos supone reducir el subintervalo de integración tanto como deseamos. El cálculo computacional se incrementa pero el error intrínseco de la integral definida (superficie bajo la curva) se hace tan pequeño como queramos.

b2) Resolución numérica de ecuaciones diferenciales (ODE, Ordinary Differential Equation)

Tal y como hemos señalado las ecuaciones diferenciales [III.4.32] y [III.4.34] no tienen solución exacta por lo que hay que recurrir a otros tratamientos como son la integración numérica o bien la resolución numérica de ecuaciones diferenciales (ODE). Existen numerosos métodos para la resolución numérica de ecuaciones diferenciales ⁽¹³⁸⁾, método de Euler "Predictor-Corrector", métodos de Runge-Kutta, método de Gear, etc., tal y como hemos explicado en el apartado III.3.2.

III.4.2.2.- Determinación de ΔH^\ddagger y ΔS^\ddagger . Ecuación de Eyring

Si consideramos la variación de k con T para una reacción química dada por la ecuación de Eyring:

$$k(T) = \frac{k_B T}{h} e^{\frac{-\Delta H^\ddagger}{RT}} e^{\frac{\Delta S^\ddagger}{R}} \quad \text{[III.4.38]}$$

k = constante de velocidad de la reacción.

T = temperatura absoluta.

R = constante de los gases ideales.

k_B = constante de Boltzman

h = constante de Planck

ΔH^\ddagger = Entalpía de activación

ΔS^\ddagger = Entropía de activación

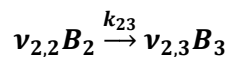
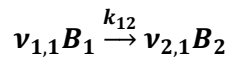
Sustituyendo en la ecuación [III.4.23] se obtiene la ecuación diferencial:

$$-\frac{1}{|v_{1,1}|} \ln \alpha_1 = \int_0^t \left(\frac{k_B}{h}\right) T e^{\frac{-\Delta H_{12}^\ddagger}{RT}} e^{\frac{\Delta S_{12}^\ddagger}{R}} dt \quad \text{[III.4.39]}$$

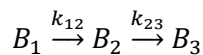
La solución de la integral del segundo miembro de la ecuación [III.4.38] depende de la función explícita $T = f(t)$ que se utiliza. Descartamos la función inversa de la temperatura correspondiente a una rama hiperbólica [III.4.27] porque en este caso no posee una solución matemática exacta de la integral. Por lo tanto, consideramos exclusivamente la función polinómica de grado n [III.4.33] y el tratamiento a realizar es idéntico que en el caso de la ecuación de Arrhenius. Consideramos 2 opciones: la integración numérica de la ecuación usando fórmulas de cuadratura apropiadas o bien realizando la resolución numérica de la ecuación diferencial [III.4.39] mediante la aplicación de métodos adecuados para el tratamiento de sistemas rígidos ordinarios (ODE).

$$d[B_1] = -[B_1] \left(\frac{k_B}{h}\right) T e^{\frac{-\Delta H_{12}^\ddagger}{RT}} e^{\frac{\Delta S_{12}^\ddagger}{R}} dt \quad \text{[III.4.39]}$$

Mecanismo II



Si consideramos $v_{1,1} = -1, v_{2,1} = 1, v_{2,2} = -1, v_{2,3} = 1$, podemos representar este mecanismo de la siguiente manera:



Acordando lo anterior, el sistema de ecuaciones diferenciales en forma matricial puede ser expresado como:

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12}(T) & 0 & 0 \\ k_{12}(T) & -k_{23}(T) & 0 \\ 0 & k_{23}(T) & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \quad \text{[III.4.40]}$$

El sistema de ecuaciones diferenciales para este sistema viene dado por:

$$\frac{d[B_1]}{dt} = -k_{12}(T)[B_1] \quad \text{[III.4.40]}$$

$$\frac{d[B_2]}{dt} = k_{12}(T)[B_1] - k_{23}(T)[B_2] \quad \text{[III.4.41]}$$

$$\frac{d[B_3]}{dt} = k_{23}(T)[B_2] \quad \text{[III.4.42]}$$

La resolución analítica del sistema de ecuaciones diferenciales en condiciones isotérmicas conduce a las soluciones:

$$[B_1] = [B_1]_0 e^{-k_{12}t} \quad \text{[III.4.43]}$$

$$[B_2] = \frac{[B_1]_0 k_{12}}{k_{23} - k_{12}} (e^{-k_{12}t} - e^{-k_{23}t}) \quad \text{[III.4.44]}$$

$$[B_3] = [B_1]_0 \left(1 - \frac{k_{23}}{k_{23} - k_{12}} e^{-k_{12}t} + \frac{k_{12}}{k_{23} - k_{12}} e^{-k_{23}t} \right) \quad \text{[III.4.45]}$$

En caso de que el proceso se realice en condiciones no isotérmicas y si el objetivo es la determinación de los parámetros termodinámicos de activación, es necesario sustituir $k_{12}(T)$ y $k_{23}(T)$ en las ecuaciones [III.4.40], [III.4.41] y [III.4.42] en función de las ecuaciones de Arrhenius y Eyring. El tratamiento de la ecuación [III.4.40] ha sido considerado en el estudio del sistema $B_1 \xrightarrow{k_{12}} B_2$ siendo en este caso el procedimiento análogo. La introducción de $k_{12}(T)$ y $k_{23}(T)$ en las ecuaciones [III.4.41] y [III.4.42] en función de los parámetros de las ecuaciones de Arrhenius y de Eyring, conduce a un sistema de ecuaciones diferenciales que carece de solución matemática exacta. Por tanto existen funciones no explícitas de $\alpha_2(t_i)$ y $\alpha_3(t_i)$ que dependen de los parámetros termodinámicos de activación y del tiempo. Las soluciones de las ecuaciones diferenciales [III.4.41] y [III.4.42] son exclusivamente numéricas, lo que significa que sólo obtendremos valores discretos de $\alpha_2(t_i)$ y $\alpha_3(t_i)$ para cada valor de tiempo. Por esa razón, es necesario determinar $k_{12}(T_i)$ y $k_{23}(T_i)$ para cada valor de T_i , sustituyendo los valores de t_i en las siguientes ecuaciones

$$k_{12}(T_i) = A_{12} e^{-E_{12}/R \sum_{j=0}^{j=n} a_j t_i^j} = \left(\frac{k_B}{h} \right) \sum_{j=0}^{j=n} a_j t_i^j e^{-\frac{\Delta H_{12}^\ddagger}{R} \sum_{j=0}^{j=n} a_j t_i^j e^{\frac{\Delta S_{12}^\ddagger}{R}}} \quad \text{[III.4.46]}$$

$$k_{23}(T_i) = A_{23} e^{-E_{23}/R \sum_{j=0}^{j=n} a_j t_i^j} = \left(\frac{k_B}{h} \right) \sum_{j=0}^{j=n} a_j t_i^j e^{-\frac{\Delta H_{23}^\ddagger}{R} \sum_{j=0}^{j=n} a_j t_i^j e^{\frac{\Delta S_{23}^\ddagger}{R}}} \quad \text{[III.4.47]}$$

Es importante tener en cuenta que los valores de A_{12} , E_{12} , ΔH_{12}^\ddagger y ΔS_{12}^\ddagger se pueden determinar previamente ya que estos valores están involucrados en la ecuación [III.4.46]. Deben mantenerse constantes en el posterior proceso de determinación y optimización de A_{23} , E_{23} , ΔH_{23}^\ddagger y ΔS_{23}^\ddagger por medio de la aplicación completa de AH a la ecuación [III.4.47].

Finalmente los valores numéricos de $\alpha_2(t_i)$ y $\alpha_3(t_i)$ se determinan por las ecuaciones [III.4.48] y [III.4.49].

$$\alpha_2(t_i) = \frac{k_{12}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} (e^{-k_{12}(T_i)t_i} - e^{-k_{23}(T_i)t_i}) \quad \text{[III.4.48]}$$

$$\alpha_3(t_i) = \left(1 - \frac{k_{23}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} e^{-k_{12}(T_i)t_i} + \frac{k_{12}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} e^{-k_{23}(T_i)t_i}\right) \quad \text{[III.4.49]}$$

Estos dos conjuntos de pares de valores cinéticos $[\alpha_2(t_i)/t_i]$ y $[\alpha_3(t_i)/t_i]$ constituyen los datos para la aplicación del AH (ANN-AGDC), lo que nos va a permitir determinar los valores de los 8 parámetros termodinámicos de activación (PTA) simultáneamente (A_{12} , E_{12} , ΔH_{12}^\ddagger , ΔS_{12}^\ddagger , A_{23} , E_{23} , ΔH_{23}^\ddagger y ΔS_{23}^\ddagger).

*El sabio no dice nunca todo lo que piensa,
pero siempre piensa todo lo que dice.
Aristóteles*

IV. Aspectos Computacionales

Capítulo IV.

Aspectos Computacionales

A lo largo de este capítulo se exponen los aspectos Computacionales que nos han permitido aplicar las diferentes metodologías (ANN, AGDC y la combinación de ambas, AH) para el tratamiento computacional de diferentes sistemas químico-físicos. Los objetivos que se han pretendido conseguir con la aplicación de estas metodologías son la determinación de Parámetros Cinéticos (*constantes de velocidad*) y Parámetros Termodinámicos de Activación (A , E , ΔH^\ddagger , ΔS^\ddagger , ΔG^\ddagger), así como la posibilidad de discriminar entre diferentes mecanismos de reacción. El Algoritmo Híbrido propuesto para alcanzar estos objetivos combina dos metodologías ANN y AGDC, cuyos fundamentos teóricos hemos consignado en el capítulo anterior por lo que a continuación expondremos los procedimientos computacionales que nos han permitido la aplicación de estos métodos.

IV.1.- Diseño de ANN. Software informático

Para diseñar una Red Neuronal en orden a resolver un problema concreto, es conveniente disponer de un software apropiado, ya que de este modo no es necesaria la programación computacional de algoritmos en lenguajes de alto nivel. La existencia de este tipo de software permite centrarse en el diseño de la Arquitectura o Estructura de la Red, en la selección de los datos del conjunto de *Entrenamiento*, en la definición de los ficheros o matrices de datos de entrada y salida y en la elección de los parámetros para los cálculos internos de la Red en el caso de que así se requieran. Elegido el software apropiado, se construye la Red especificando el número de capas y de neuronas en cada capa, los tipos de conexiones y su topología. Además se pueden seleccionar diferentes *funciones de transferencia* (funciones “Delta”) y procesamiento de las neuronas. En la actualidad existe en el mercado un software comercial para el diseño y aplicación de ANN a muy diversos problemas. Los que se utilizan con mayor profusión por parte de la comunidad científica son, MATLAB⁽⁶⁴⁾ (el más usado) y en menor medida, TRAJAN⁽⁶¹⁾. Nosotros hemos utilizado las aplicaciones proporcionadas por **MATLAB** (abreviatura de *MATrix LABORatory*), el cual es un software matemático con un lenguaje de programación propio (lenguaje M y extensiones ejecutables “.m”). Entre sus prestaciones básicas más utilizadas en este trabajo se hallan: Resolución de sistemas de ecuaciones diferenciales (ODE) e integrales usando el cálculo simbólico y numérico, Álgebra y operaciones con matrices, representación de datos y funciones (2D y 3D), implementación de algoritmos, creación de *interfaces* de usuario (GUI), comunicaciones y ensamblajes con programas en otros lenguajes (EXCEL, MATHEMATICA, C++,..., etc.) y con otros

dispositivos hardware. MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, SIMULINK (plataforma de simulación *multidominio*) y GUIDE (editor de *interfaces* de usuario - GUI). Además posee una serie de aplicaciones agrupadas en *toolboxes* que permiten alcanzar diferentes objetivos, dentro de estos grupos destacamos *Optimization ToolboxTM*, *Partial Differential Equation ToolboxTM* y *Neural Network ToolboxTM*. Este último conjunto de aplicaciones es el que hemos utilizado a lo largo del trabajo que recoge la presente Memoria para el diseño y aplicación de ANN, en el tratamiento de datos para determinar los diversos parámetros.

Neural Network ToolboxTM posibilita la determinación de parámetros (*Predicción*) además de la realización de funciones de clasificación. Para ello dispone de diferentes aplicaciones que realizan las siguientes tareas:

- *Entrenamiento* de una Red neuronal para ajustar una función.
- *Entrenamiento* de una Red neuronal para reconocer patrones.
- *Entrenamiento* de una Red neuronal para agrupar datos.

Puesto que nuestro objetivo es la determinación de parámetros cinéticos y termodinámicos a partir de datos experimentales, utilizaremos la aplicación que permite ajustar funciones ("*Fitting a Function*"), cuyo funcionamiento explicaremos a continuación. Supongamos que disponemos de datos correspondientes a una serie de experiencias cinéticas, realizadas bajo diferentes condiciones, junto con los correspondientes valores de las constantes de velocidad o Parámetros Termodinámicos de Activación y pretendemos determinar el valor de los mismos en otras experiencias realizadas en condiciones diferentes mediante el diseño de una Red Neuronal que nos resuelva el problema. Disponemos de 2 opciones: Utilizar Comandos (*Command-Line Functions*) o bien hacer uso de la *interfaz* gráfica (*nftool*). Mediante cualquiera de estas aplicaciones podemos llevar a cabo la *Predicción* de los parámetros deseados, para lo cual es necesario realizar una serie de pasos que detallamos a continuación.

IV.1.1.- Ajuste de funciones en ANN

En un problema de ajuste a funciones matemáticas con *Neural Network ToolboxTM*, es necesario disponer del conjunto de datos que se van a utilizar para el *Entrenamiento* de la Red (*Training set*). Estos datos son proporcionados a la Red Neuronal Artificial (ANN) y contienen matrices de entrada (*Inputs*) junto con las correspondientes salidas (*Outputs*) y ésta "*aprende*" a establecer relaciones entre los dos. Para el correcto funcionamiento de la aplicación, es necesario preparar "*a priori*" una matriz de entrada (*Inputs*) y una matriz de *Targets*, constituida por los parámetros correctos correspondientes a los datos de la matriz de entrada (*Inputs*). Estos datos son sintéticamente generados para cada mecanismo de reacción considerado y hay que

organizarlos de una forma apropiada en orden a suministrárselos a la ANN de acuerdo a las exigencias que impone la aplicación de MATLAB.

IV.1.1.1.- Generación de datos

Hemos estudiado la aplicación de ANN para la determinación de las constantes de velocidad y los Parámetros Termodinámicos de Activación de varios sistemas, para lo que es necesario generar datos sintéticos de concentración/tiempo, absorbancia/tiempo, y/o fracción remanente/tiempo a partir de los parámetros que proporciona el Diseño de Experimentos (ED). Estos datos se generan mediante un programa ejecutable (tipo “.m”) de MATLAB, diseñado y ejecutado en nuestro laboratorio, teniendo en cuenta las condiciones iniciales de concentración, los coeficientes de absorción molar, la variación de la temperatura, y el intervalo de tiempo en el que se llevan a cabo las reacciones de acuerdo a lo previsto en el correspondiente Diseño de Experimentos.

IV.1.1.1.a.- Datos concentración/tiempo

Los datos $[B_j]/t_i$ se generan computacionalmente mediante un programa ejecutable MATLAB del tipo (“.m”), diseñado y ejecutado por nosotros aplicando las ecuaciones resultantes de la resolución exacta (en los casos posibles) de los sistemas de ecuaciones diferenciales de velocidad correspondientes a los mecanismos de reacción que estamos considerando.

IV.1.1.1.b.- Datos absorbancia/tiempo

Una vez que se han calculado los datos de concentración, se determinan de igual manera, los datos de absorbancia de cada una de las especies químicas B_j al tiempo t_i y la longitud de onda λ ($A_{j,t_i}^\lambda/t_i$). Los datos generados se distribuyen de la siguiente manera en las matrices de *Inputs* y *Targets* que tienen las siguientes dimensiones:

Matriz de *Inputs* (N° filas x N° columnas)

N° filas = N° datos de cada experiencia

N° columnas = N° de curvas

Matriz de *Targets* (N° filas x N° columnas)

N° filas = N° de parámetros

N° columnas = N° de curvas

IV.1.1.1.c.- Datos fracción remanente/tiempo

Como ya se ha mencionado anteriormente es necesario generar datos de la fracción remanente α a partir de los valores de los parámetros de activación, E (energía de activación), A

(factor preexponencial), entropía de activación ΔS^\ddagger y/o entalpía de activación ΔH^\ddagger que nos proporciona el Diseño de Experimentos previamente realizado. Estos datos se generan mediante un programa ejecutable que hemos diseñado en el Editor de MATLAB.

En otros casos se ha llevado a cabo la resolución numérica del sistema de ecuaciones diferenciales (ODE) mediante las herramientas proporcionadas por MATLAB, aplicando distintas metodologías numéricas de resolución. Tal y como hemos mencionado en el capítulo anterior, existen numerosos métodos para la resolución ODE's: Euler, Runge-Kutta de diferentes órdenes, Gear, Adams, etc... De acuerdo con las características del sistema a integrar es conveniente la utilización de un método u otro. MATLAB proporciona "*funciones*" para la integración de ODE, que mostramos en la siguiente tabla, en la que destacamos algunos de ellos con sus características más relevantes en cuanto a exactitud, el tipo de sistema para el que está indicado y el algoritmo que utiliza.

Método	Tipo sistema	Exactitud	Algoritmo
Ode45	No stiff	Media	Runge-Kutta
Ode23	No stiff	Baja	Runge-Kutta
Ode113	No stiff	Baja-Alta	Adams
Ode15s	stiff	Baja-Media	Gear

Tabla IV.1: Diferentes métodos para la integración de ODE con su tipo de sistema, exactitud y algoritmo.

Estos métodos de resolución de ODE's han sido aplicados para nuestro caso particular y hemos observado que los métodos más exactos son ode45 y ode113, ambos dan los mismos resultados y puesto que MATLAB recomienda ode45 como preferente, utilizaremos este método como el más apropiado.

IV.1.1.2.- Representaciones gráficas

La aplicación de la metodología ANN en Cinética Química implica trabajar con un colectivo muy numeroso de datos donde la representación gráfica de los mismos puede ayudarnos a visualizar el proceso. Existen numerosas aplicaciones informáticas que permiten realizar representaciones gráficas, MATLAB, ORIGIN, EXCEL, QTILOT, etc... En este caso hemos utilizado MATLAB para la representación en 3D de todas las curvas generadas absorbancia total/tiempo ($A_{T,t_i}^\lambda/t_i$), absorbancia individual/tiempo ($A_{j,t_i}^\lambda/t_i$), fracción remanente individual/tiempo ($\alpha_{j,t_i}/t_i$), y concentración/tiempo ($[B_j]/t_i$). Hemos desarrollado una amplia casuística en las representaciones gráficas 2D Y 3D que hemos trazado en el presente trabajo y cuyo tratamiento merece un apartado específico. En él, exponemos en detalle la organización matricial de datos, las técnicas de representación, las características del software comercial y "libre" utilizado, así como un amplio colectivo de figuras representativas de un extenso número de casos y situaciones.

IV.1.2.- Diseño de la arquitectura de la ANN

La Red Neuronal tiene que ser diseñada para cada sistema concreto lo que significa buscar la *Arquitectura* óptima, es decir, el número óptimo de capas (Estructura) y el número de nodos en cada capa (Configuración). Para ello debemos crear la Red indicando en primer lugar su Estructura y a continuación la Configuración, es decir, el número de nodos en cada una de las capas ocultas (*Hidden*), procediendo de la siguiente manera:

- a) Diseño de una Red constituida por 2 capas (1 *Hidden* + 1 *Output*)
 - i) Establecer el número de neuronas de la capa *Hidden* ($numHiddenNeurons = N^{\circ} \text{ entero}$)
 - ii) Crear la Red ($net = newfit(inputs, targets, numHiddenNeurons)$)
- b) Diseño de una Red formada por n capas ((n-1) *Hidden* + 1 *Output*)
 - i) Establecer el número de capas ($numLayer=n$)
 - ii) Establecer el número de neuronas de las capas ocultas (*Hidden*) ($numHiddenNeurons1 = N^{\circ} \text{ entero}, numHiddenNeurons2 = N^{\circ} \text{ entero}, \dots$)
 - iii) Crear la Red ($net = newfit(inputs, targets, [numHiddenNeurons1, numHiddenNeurons2, \dots])$)

IV.1.3.- Entrenamiento o Aprendizaje de la Red Neuronal Artificial

La ANN tiene que ser sometida a una etapa de *Aprendizaje* o *Entrenamiento* para que funcione correctamente y de este modo obtener los resultados deseados. El *Entrenamiento* se lleva a cabo con los datos correspondientes al *Training data set* que se dividen en tres grupos en los siguientes porcentajes:

- 60% de las experiencias se utilizan para el *Entrenamiento*.
- 20% se utilizan para validar la Red (generalización) y parar el proceso antes de que se produzca *overfitting*.
- 20% se utiliza como una prueba independiente, *Testing*.

Estos porcentajes pueden ser modificados en cualquier momento por el usuario para mejorar los resultados. Para llevar a cabo el *Entrenamiento*, MATLAB aplica por defecto el algoritmo Levenberg-Marquardt entre una amplia serie de opciones que posee. El proceso de *Entrenamiento* se lleva a cabo siguiendo los pasos que se señalan a continuación:

- a) **Entrenamiento** de la ANN (*Training*) ($net=train(net, inputs, targets)$)

Mientras se lleva a cabo el *Entrenamiento* se puede visualizar la evolución del proceso iterativo de optimización que tiene lugar y que puede ser detenido si se produce un aumento en el error de *Validación*. En todo caso, se pueden ver los errores de *Entrenamiento*, de *Validation* y *Testing* final. Cuando ha finalizado el proceso iterativo aparecen una serie de parámetros (MSE general, Regresión entre valores de *Outputs* y *Targets*, número de iteraciones ("*Epochs*"),

"Performance", valor del vector Gradiente del algoritmo Levenberg-Marquardt (valor de μ), número de chequeos de *Validación...etc*) que nos permitirán deducir si el *Entrenamiento* de la ANN se ha llevado a cabo con éxito o no. El resultado es aceptable si se cumplen los siguientes requisitos:

- El error cuadrático medio (MSE) es significativamente pequeño.
- Los errores de *Validación* y *Testing* tienen características similares.
- No se produce *overfitting*.
- Evaluación satisfactoria de la respuesta de la ANN, realizando una regresión lineal entre las correspondientes *Outputs* (salidas) de la Red y los *Targets*.

El análisis de todos estos factores permite decidir si el *Entrenamiento* ha sido realizado satisfactoriamente. Si es así la Red está lista para realizar tareas de *Predicción*, y en caso contrario podemos mejorar los resultados teniendo en cuenta las siguientes consideraciones:

b) Mejora de los resultados:

- Restablecer la ANN inicial y entrenarla de nuevo, "*Retrain*".
- Aumentar el número de Neuronas de las capas *Hidden* (distinta Configuración).
- Aumentar el número de capas *Hidden*.
- Aumentar el número de *Inputs* si se dispone de más información.
- Aumentar el número de vectores de *Entrenamiento*, modificando el porcentaje de experiencias para "*Training*", "*Test*" y "*Validation*".
- Utilizar otro algoritmo de optimización más robusto que minimice las diferencias de los valores de las matrices de *Outputs* y *Targets*.

IV.1.4.- *Predicción*

La Red óptima se puede utilizar para diversas tareas de *Predicción* como es la determinación de parámetros desconocidos. Para llevar a cabo el proceso de *Predicción* se deben realizar las siguientes etapas:

1.- Proporcionar los datos a la ANN: hay que introducir los datos correspondientes a las experiencias cuyos parámetros son desconocidos:

Matriz de *Inputs* (Nº filas x Nº columnas)

Nº filas = Nº datos de cada experiencia, que debe coincidir con el nº de filas de la matriz *Inputs* con la que ha sido entrenada la ANN.

Nº columnas = Nº de parámetros desconocidos.

2.- *Predicción*: a partir de la Red Neuronal óptima previamente entrenada, para obtener el vector que contiene los valores de los parámetros desconocidos y que se agrupan en la matriz [Y] mediante el comando:

$[Y]=sim(net,inputs)$

IV.1.5.- Procedimiento Computacional de MATLAB. Forma algorítmica

El procedimiento computacional que utiliza MATLAB, correspondiente a los procesos de *Entrenamiento* o *Aprendizaje y Predicción* que hemos explicado en los Apartados IV.1.2 a IV.1.4, puede compendiarse de una manera esquemática en forma algorítmica, de la siguiente manera:

A.- INICIO del proceso de *Entrenamiento* de la Red Neuronal

1.- Introducir matrices de las curvas sintéticas en la Capa de Entrada (*Input Layer*):

- a) *Inputs*: $(A_{T,t_i,n_c}^\lambda; A_{j,t_i,n_c}^\lambda; [B_j]_{t_i,n_c}; \alpha_{j,t_i,n_c})$
 b) *Targets*: $(k_{12}, k_{23}; k_{12}, k_{23}, k_{31}; k_{12}, k_{21}, k_{23}; k_{12}, k_{23}, k_{32}; A_{12}, E_{12}; A_{23}, E_{23}; \Delta H_{12}^\#, \Delta S_{12}^\#; \Delta H_{23}^\#, \Delta S_{23}^\#)$

2.- Elegir o Modificar (en caso de mejora de resultados)

- a) Número de capas Ocultas (*Hidden*)(Estructura)
 b) Número de neuronas (nodos) por capa/s oculta/s (Configuración de capa/s)
 c) Porcentajes de: *Training* (TR), *Validation* (VL) y *Testing* (TS)
 d) Algoritmo de Optimización *Outputs* respecto a *Targets*.
 d1) *Levenberg Marquardt* (LM) (elección por defecto)
 d2) " *Quasi-Newton*" (QN)
 d3) *Gradientes Conjugados* (GC)
 d4) *Steepest Descent* (SD).....etc.

3.- Entrenamiento. Control de:

- a) Número de iteraciones (*epochs*) (< 1000)
 b) Tiempo transcurrido (divergencia) *Testing* (TS)
 c) *Performance*: Perfiles curvas MSE (*Mean Squared Error*) (TR, VL y TS)
 d) Estado del proceso de Entrenamiento (*Training State*)
 → Análisis de los perfiles de las gráficas (MSE vs número de iteraciones (*epochs*)) de:
 d1) superposición de curvas de TR, VL y TS
 d2) vector Gradiente algoritmo Optimización (módulo ~ 0 vector movimiento)
 d3) μ vs número de iteraciones (*epochs*)
 d4) *Validación* → nº veces aumento MSE (≤ 5 ; *epoch* del fracaso de *Validación*)
 d5) chequeos de *Testing* (MSE vs. tiempo)
 e) Existencia de *Overtraining*
 e1) no se detecta superposición de curvas de MSE (TR, VL y TS)

4.- Evaluación de Resultados (*Outputs*)

- a) MSE, (TR, VL, TS) → *Performance*: mínimo valor MSE (*Best Validation*)
 b) Regresión *Outputs* / *Targets* (TR, VL, TS)
 b1) Ecuaciones lineales (pendiente ~ 1, ordenada ~ 0)
 b2) R (Coeficiente de Regresión) ~ 1
 b3) Representaciones gráficas → Dispersión
 b4) Curvas *Performance* vs. número de *epochs* (TR, VL, TS) → *Best Validation*
 b5) Curvas *Performance* vs. tiempo (TR, VL, TS) → *Testing*
 c) Valores finales
 c1) Módulo vector, Gradiente algoritmo de optimización (LM, QN, GC, SD...) → Evaluación avance vector movimiento)
 c2) Tiempo total *Testing* (posible divergencia)

- c3) Número total de iteraciones (*epochs*)
 - c4) *Overtraining* (alejamiento de la curva Training a valores menores de MSE)
 - c5) μ (*momentum constant*) → control del módulo vector de movimiento (LM)
 - d) Ajuste (TR, VL, TS)
 - d1) Superposición puntos, *Output and Targets* (TR, VL, TS)
 - d2) Líneas (rectas) de Regresión (TR, VL, TS)
 - e) Arquitectura óptima de la ANN *entrenada* (óptima-net)
- 5.- Análisis de Residuales. Cálculo de:

- a) datos cinéticos recalculados a partir de *outputs*

$$(A_{T,t_i,n_c}^{(calc)}; A_{j,t_i,n_c}^{(calc)}; [B_j]_{t_i,n_c}^{(calc)}; \alpha_{j,t_i,n_c}^{(calc)})$$

- b) función Suma Desviaciones Cuadráticas (*SQD*)

$$SQD = \sum_{i=1}^{t_{\infty}} (A_i^{(calc)} - A_i^{(input)})^2$$

$$SQD = \sum_{i=1}^{t_{\infty}} (\alpha_i^{(calc)} - \alpha_i^{(input)})^2$$

- c) Varianza, Desviación Estándar (*SD*) y Error

$$SD = \left(\frac{\sum_{i=1}^{t_{\infty}} (A_i^{(calc)} - A_i^{(input)})^2}{n_d} \right)^{1/2}$$

$$SD = \left(\frac{\sum_{i=1}^{t_{\infty}} (\alpha_i^{(calc)} - \alpha_i^{(input)})^2}{n_d} \right)^{1/2} \quad n_d = \text{número de datos}$$

- d) Matriz de Error
- e) Gráfica de distribución Residuales vs. t_i

6.- Decisión respecto a resultados: posible Mejora de los mismos

- a) Valores NO satisfactorios de los parámetros:
 - a1) si se desea añadir/eliminar valores matriz *Inputs*, ir a la etapa 1.
 - a2) si se desea modificar las variables elegidas en 2, ir a la etapa 2.
- b) Valores Satisfactorios: Guardar:
 - b1) Arquitectura óptima (*net* óptima): Capas ocultas y Nodos/capa (Configuración)
 - b2) MSE (TR, VL, TS)
 - b3) Matriz *Outputs*
 - b4) Matriz *Errors*
 - b5) Desviación Estándar, ... etc.

➔ (FIN del Proceso de *Entrenamiento*)

B.- INICIO del proceso de *Predicción* de la ANN

- 7.- Comando “sim”. Suministrar a la Red Neuronal (*entrenada* en etapas 1 a 6):
 - a) Arquitectura óptima de la ANN “*entrenada*” (*net* óptima)
 - b) Matriz de *Inputs* (curvas experimentales con parámetros desconocidos)
- 8.- Matriz [Y] de Resultados. Valores de los parámetros.

➔(FIN del Proceso de *Predicción*)

IV.2.- Aplicación computacional del algoritmo AGDC

La aplicación computacional del algoritmo AGDC se lleva a cabo utilizando diferentes aplicaciones informáticas, desarrolladas por nuestro grupo de investigación para su utilización en distintos entornos y para la determinación de distintos parámetros cinéticos, analíticos y termodinámicos. Por un lado hemos utilizado el programa **KINMODEL(AGDC)@Vs.1.0** ^(68,69), escrito en lenguaje FORTRAN 90 y compilado con *Microsoft Developer Studio Compiler* (©Microsoft Corporation) para su utilización en el entorno DOS. Este programa se ha utilizado para la determinación de las constantes individuales de velocidad de diferentes sistemas y la discriminación entre mecanismos de reacción. Para el caso en que se determinen Parámetros Termodinámicos de Activación (PTA), se ha desarrollado una aplicación computacional **KINNOISOT(AGDC)** basada en el algoritmo AGDC que utiliza el entorno MATLAB para su programación, aprovechando de esta manera “*funciones*” de las que dispone MATLAB para realizar diferentes cálculos numéricos necesarios para la aplicación del algoritmo AGDC.

A lo largo de este capítulo expresaremos de forma sistemática, el desarrollo y contenido de todos los programas de cálculo, mediante la exposición de los diagramas de flujo y esquemas correspondientes. De modo general, en todo proceso de optimización se determinan los valores de los parámetros desconocidos que mejor se ajustan a los datos experimentales de los que se dispone. Inicialmente se asignan unos valores a los parámetros (estimaciones iniciales) y posteriormente, a partir de los valores experimentales y los calculados con las estimaciones de los parámetros, se lleva a cabo de forma iterativa el proceso de optimización.

La Figura IV.2.1 representa un esquema del procedimiento global que se pretende realizar:

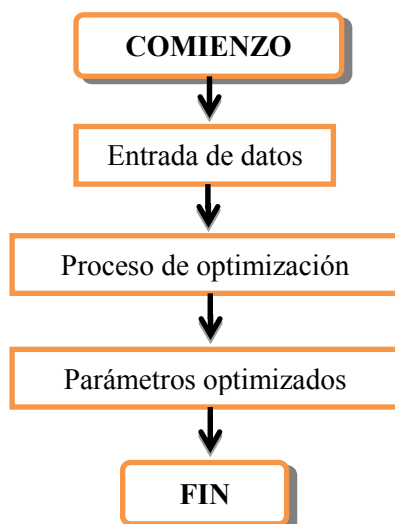


Figura IV.2.1: Esquema del procedimiento global de optimización.

Independientemente de los parámetros que se pretenden optimizar la función a minimizar es la función suma de desviaciones cuadráticas (SQD). Esta función no se conoce

explícitamente (función numérica) y su valor en cada punto puede determinarse a partir de la ecuación [III.3.4]:

$$SQD = \sum_{i=1}^{N_d} (f_i(X_1, X_2, \dots, X_p, t_i) - (y_i)_E)^2$$

donde f_i representa la función analítica que proporciona el valor teórico de la propiedad (concentración, absorbancia, fracción remanente...) con la que se monitoriza el sistema en estudio y X_1, X_2, \dots, X_p ($k_r; A_r, E_r; \Delta H_r^\ddagger, \Delta S_r^\ddagger$) son los parámetros que se pretenden determinar, siendo $(y_i)_E$ el valor experimental de la propiedad obtenida a los distintos valores de tiempo.

Inicialmente se asignan ciertos valores estimados a los parámetros que se pretenden determinar, a partir de estos valores, se calcula el valor de la propiedad del sistema utilizada en su estudio (concentración, absorbancia, fracción remanente...). A continuación se determina de forma explícita la función SQD a partir de los valores experimentales y calculados de la propiedad con la que se monitoriza el sistema en estudio. Una vez definida la función objetivo se realiza el proceso de optimización completando varias iteraciones hasta alcanzar los puntos óptimos convergentes de la optimización.

Para llevar a cabo el proceso de optimización matemática mediante el algoritmo AGDC, es necesario realizar una serie de cálculos numéricos:

- Establecimiento del sistema de ecuaciones diferenciales de velocidad.
- Resolución numérica de ecuaciones diferenciales.
- Cálculo de la función SQD .
- Cálculo de las derivadas numéricas.
- Determinación del vector gradiente y la matriz Hessiana (\mathbf{g} y \mathbf{H}).
- Inversión de la matriz Hessiana (\mathbf{H}^{-1}).
- Aplicación del algoritmo AGDC.

De manera general los programas de cálculo constan de un *programa principal* donde se lleva a cabo la optimización mediante el algoritmo AGDC y una serie de *funciones* o *subrutinas* que realizan el cálculo de las distintas partes que son necesarias para la optimización. La Figura IV.2.2 muestra el diagrama de flujo correspondiente al programa principal.

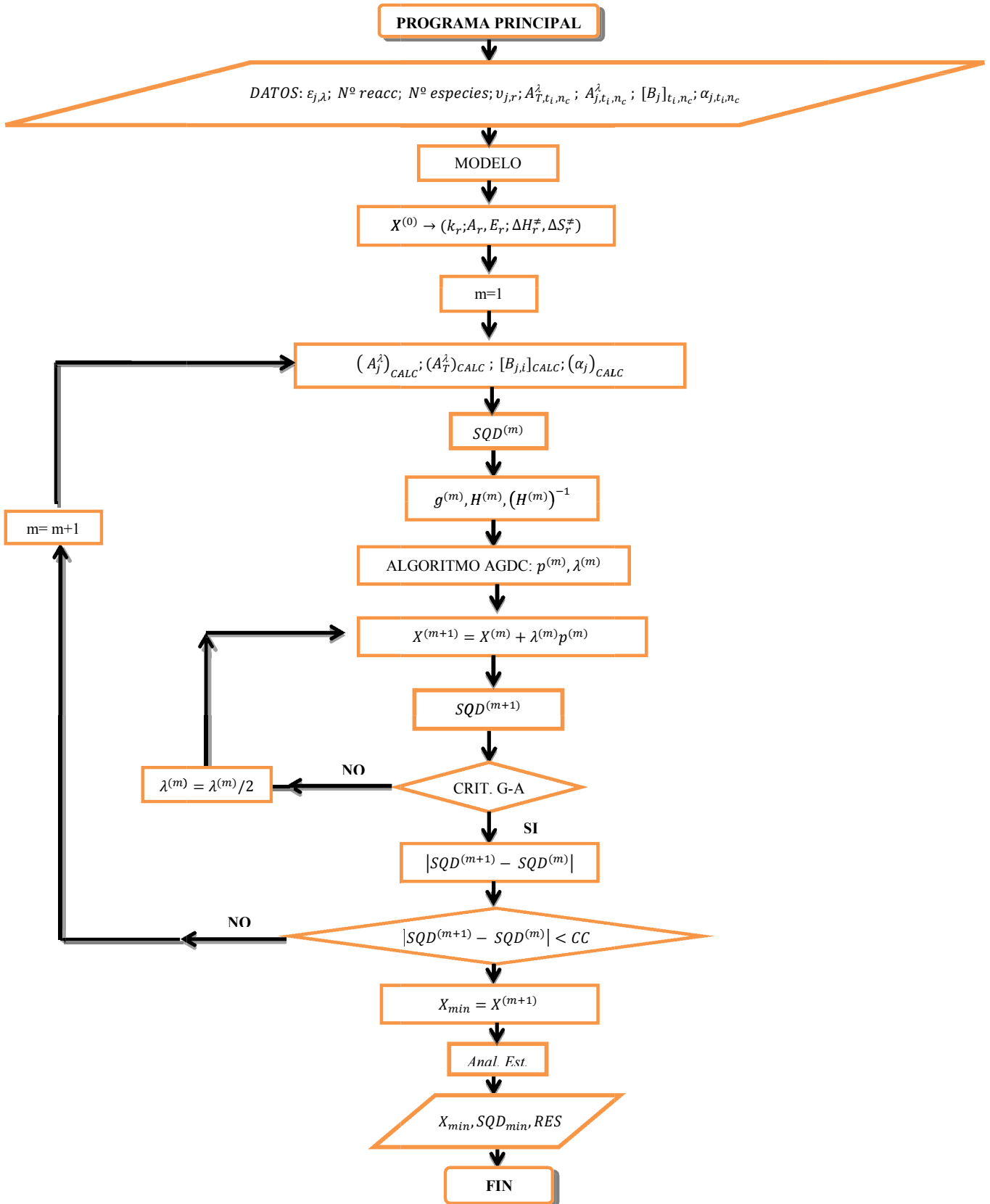


Figura IV.2.2: Diagrama de flujo del programa principal de optimización.

IV.2.1.- Optimización de constantes de velocidad

Los valores de las constantes de velocidad que han sido obtenidos mediante el proceso de *Predicción* realizado con ANN, son sometidos a un proceso de optimización matemática mediante una aplicación computacional, **KINMODEL(AGDC)@Vs.1.0** ⁽⁶⁸⁻⁶⁹⁾ que utiliza el algoritmo de optimización matemática AGDC.

KINMODEL(AGDC)@Vs.1.0 ⁽⁶⁸⁻⁶⁹⁾ es un programa de tratamiento numérico, que hemos diseñado para el estudio de cualquier modelo de reacción. Permite la determinación de diferentes parámetros cinéticos y/o analíticos, utilizando para la optimización el algoritmo AGDC. Está diseñado para la determinación, en cualquier mecanismo de reacción, de las concentraciones iniciales de las especies que intervienen, los coeficientes de absorción molar de las especies involucradas y las constantes de velocidad de las diferentes etapas del mecanismo estudiado, bien de forma independiente o conjunta. Además está capacitado para la discriminación de modelos y validación del que mejor se ajuste a los datos experimentales.

En el trabajo que recoge la presente Memoria se ha utilizado el programa **KINMODEL(AGDC)@Vs.1.0** para la determinación de las constantes de velocidad de las diferentes etapas elementales de los mecanismos considerados por lo que nos centraremos en el procedimiento computacional empleado para determinar las mismas, obviando las demás posibilidades que ofrece el programa.

La determinación de las constantes de velocidad mediante el algoritmo de optimización matemática AGDC, supone la realización de una serie de etapas que detallamos a continuación.

IV.2.1.1.- Establecimiento y resolución computacional del sistema de ecuaciones diferenciales de velocidad

El algoritmo AGDC minimiza la función suma de desviaciones cuadráticas (SQD), que no se conoce explícitamente, pero su valor en cada punto puede determinarse en este caso a partir de la ecuación [IV.2. 1]:

$$\text{SQD} = \sum_{i=1}^{N_d} \left((A_{T,t_i}^\lambda)_C - (A_{T,t_i}^\lambda)_E \right)^2 \quad \text{[IV.2. 1]}$$

donde $(A_{T,t_i}^\lambda)_E$ es el valor de la absorbancia total (T) de la mezcla reaccionante obtenido experimentalmente a cada tiempo t_i a la longitud de onda λ y $(A_{T,t_i}^\lambda)_C$ es el valor de la absorbancia calculada con los valores de los parámetros que se van determinando en cada iteración durante el proceso de optimización. Los valores de A_{T,t_i}^λ serán la suma de las absorbancias individuales de las especies j (A_{j,t_i}^λ) que intervienen en la reacción y que presentan absorción a la longitud de onda de trabajo. De acuerdo a la ley de Lambert-Beer-Bouguer, la

absorbancia de cualquier especie j al tiempo t_i y la longitud de onda λ , (A_{j,t_i}^λ), medida con un paso óptico de 1 cm, viene expresada por:

$$A_{j,t_i}^\lambda = \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad \text{[IV.2.2]}$$

siendo ε_j^λ el coeficiente de absorción molar de la especie j a la longitud de onda λ y $[B_j]_{t_i}$ la concentración molar de la especie B_j en el tiempo t_i . Por lo tanto para calcular la absorbancia es necesario realizar previamente el cálculo de las concentraciones de cada una de las especies que intervienen en la reacción, lo cual se lleva a cabo computacionalmente mediante el establecimiento y resolución numérica del sistema de ecuaciones diferenciales de velocidad. El proceso computacional para establecer el sistema genérico de ecuaciones diferenciales de velocidad obtenido a partir de la ecuación [III.4.3]:

$$d[B_j]_{t_i} = \sum_r^{N_r} k_r v_{j,r} (\prod_c^{N_c} [B_c]^{v_{c,r}}) dt$$

se realiza mediante la subrutina **DIFFEQ**, escrita en FORTRAN 90, cuya estructura se esquematiza en el diagrama de flujo indicado en la Figura IV.2.3.

Para el establecimiento del sistema de ecuaciones diferenciales de velocidad deben conocerse:

- Las concentraciones $[B_j]_{t_i}$ de todos los componentes al inicio del intervalo diferencial de tiempo dt_i en el que se van a calcular las concentraciones.
- La constante de velocidad para cada una de las reacciones (k_r).
- La matriz de coeficientes estequiométricos ($v_{j,r}$) para cada componente en cada una de las reacciones individuales.

Una vez establecido el sistema de ecuaciones diferenciales se lleva a cabo la resolución numérica del mismo mediante el algoritmo de Gear^(138,151-153). El proceso se realiza de manera computacional mediante un “paquete de subrutinas” correspondientes a la librería de programas **HARWELL**⁽¹³⁷⁾. La subrutina principal, DC03A/AD, resuelve el sistema de ecuaciones diferenciales en el intervalo de tiempo especificado por el usuario, desde T hasta TOUT, pudiéndose obtener la solución en un punto intermedio. La exactitud y precisión deseada para la integración se controla mediante el parámetro TOL.

El conjunto de subrutinas necesarias para la integración comprende todas aquellas bajo la denominación genérica DC03x/xD, donde x indica las letras A-J Y M-X, además de las subrutinas correspondientes a la librería **HARWELL**: MA33A/AD, MA33B/BD, MA34A/AD, MA34B/BD, MA30A/AD, MA30B/BD y MA30C/CD. Hemos agrupado todas las subrutinas necesarias, en la denominada **RESECDF** que a su vez se une con la subrutina **DIFFEQ** en el bloque **GEARHAR**, facilitando de este modo la evaluación de las concentraciones en el programa general de cálculo. El diagrama de flujo se muestra en la Figura IV.2.4.

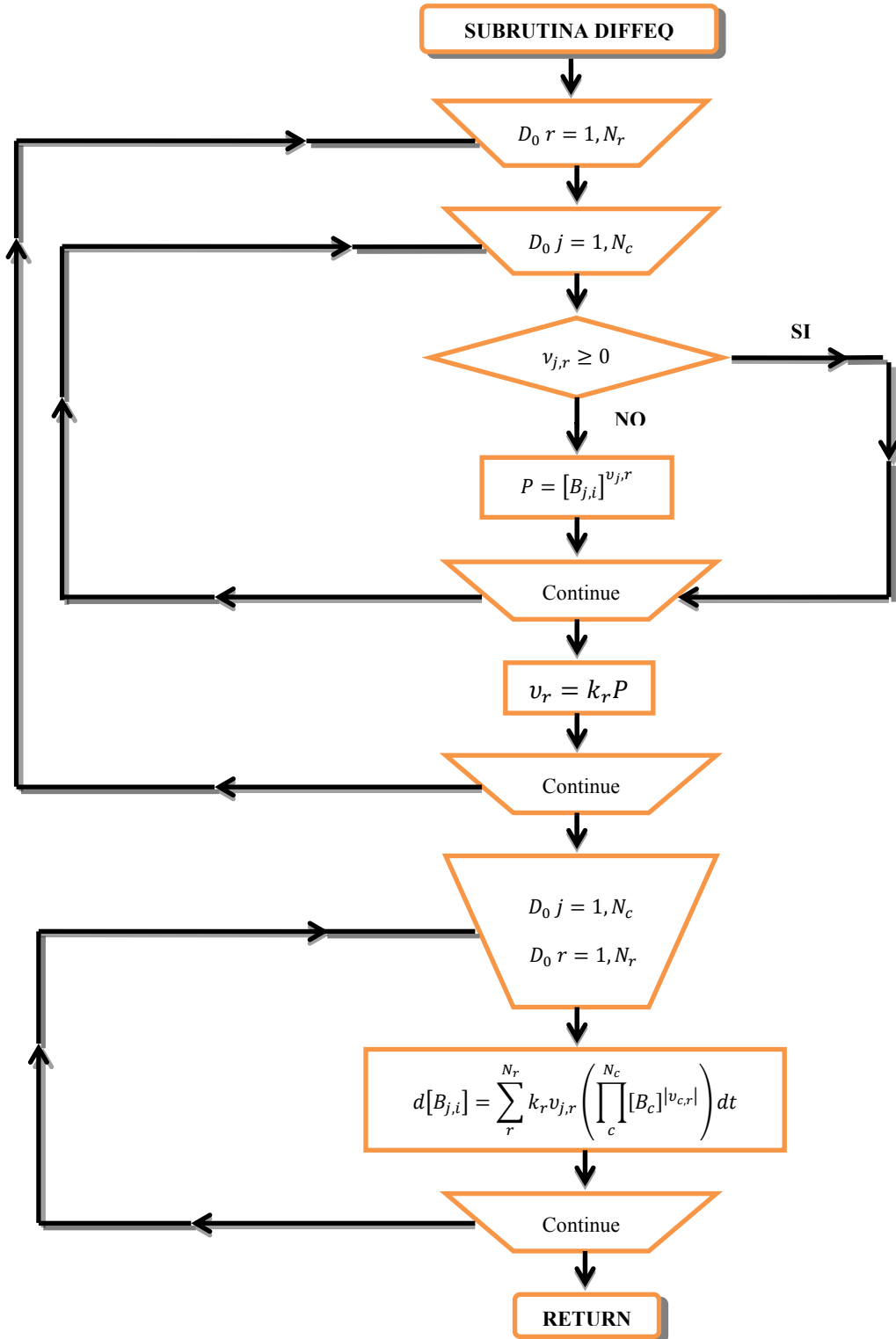


Figura IV.2.3: Diagrama de flujo de la subrutina DIFFEQ.

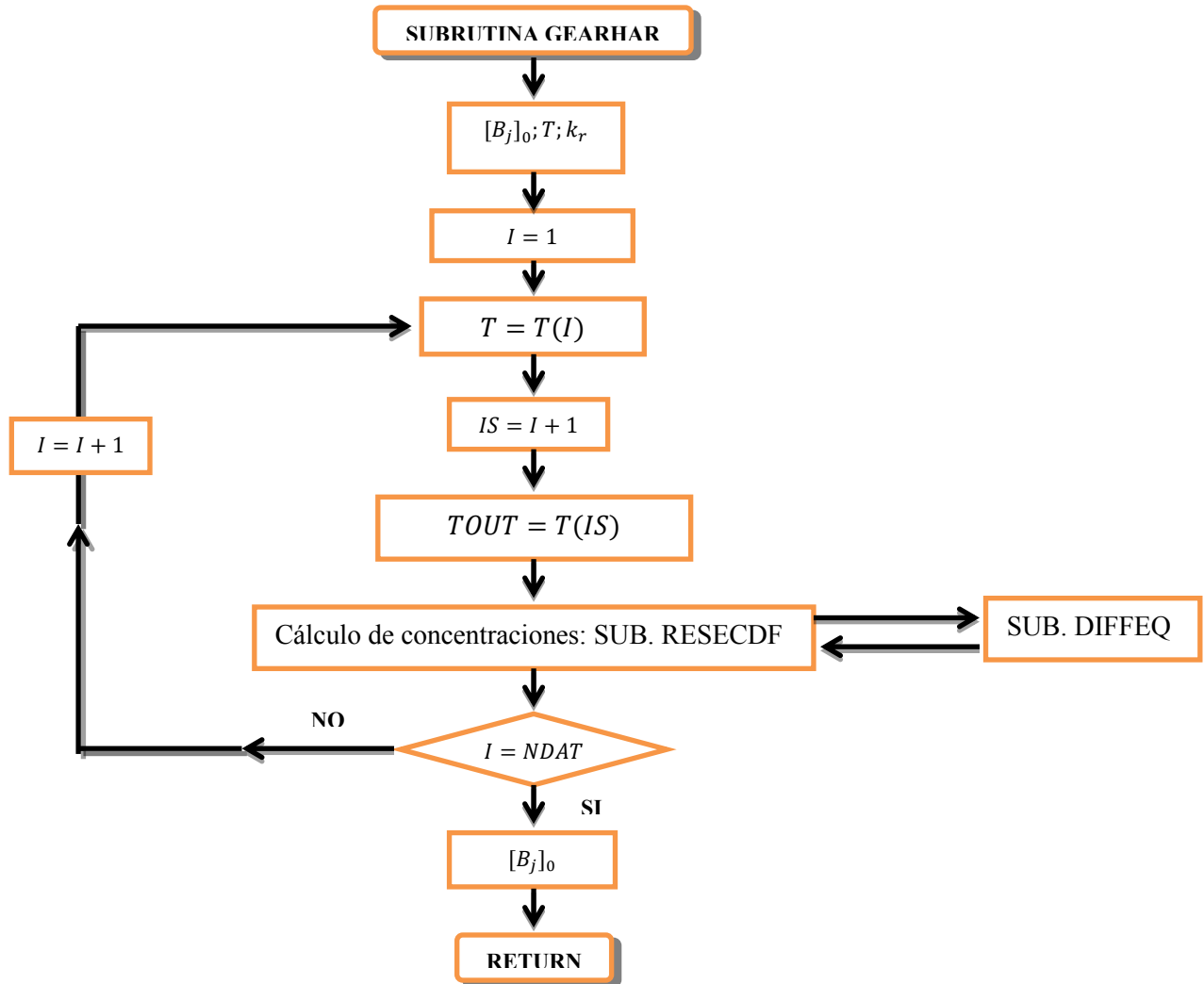


Figura IV.2.4: Diagrama de flujo de la subrutina GEARHAR.

IV.2.1.2.- Cálculo computacional de las derivadas numéricas

La aplicación del algoritmo AGDC en el proceso de optimización conlleva la determinación del vector gradiente y la matriz Hessiana, para el cálculo del vector de movimiento de acuerdo con la ecuación [III.3.24]:

$$\mathbf{p}^{(m)} = \Delta \mathbf{X}^{(m)} = -\mathbf{H}(\mathbf{X}^{(m)})^{-1} \mathbf{g}(\mathbf{X}^{(m)})$$

Las ecuaciones [III.3.11] y [III.3.15] proporcionan la expresión general de \mathbf{g} y \mathbf{H} . Cuando los parámetros a optimizar son las constantes de velocidad obtenemos las siguientes expresiones:

$$\mathbf{g} = 2 \begin{bmatrix} \sum_1^{Nd} RES_i \frac{\partial A_i}{\partial k_1} \\ \sum_1^{Nd} RES_i \frac{\partial A_i}{\partial k_2} \\ \sum_1^{Nd} RES_i \frac{\partial A_i}{\partial k_3} \\ \vdots \\ \sum_1^{Nd} RES_i \frac{\partial A_i}{\partial k_r} \end{bmatrix} \quad [\text{IV.2.3}]$$

$$\mathbf{H} = 2 \begin{bmatrix} \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_1} \right)^2 & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_1} \frac{\partial A_i}{\partial k_2} \right) & \dots & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_1} \frac{\partial A_i}{\partial k_r} \right) \\ \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_2} \frac{\partial A_i}{\partial k_1} \right) & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_2} \right)^2 & \dots & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_2} \frac{\partial A_i}{\partial k_r} \right) \\ \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_3} \frac{\partial A_i}{\partial k_1} \right) & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_3} \frac{\partial A_i}{\partial k_2} \right) & \dots & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_3} \frac{\partial A_i}{\partial k_r} \right) \\ \vdots & \vdots & \dots & \vdots \\ \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_r} \frac{\partial A_i}{\partial k_1} \right) & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_r} \frac{\partial A_i}{\partial k_2} \right) & \dots & \sum_1^{Nd} \left(\frac{\partial A_i}{\partial k_r} \right)^2 \end{bmatrix} \quad [\text{IV.2.4}]$$

$$RES_i = ((A_i)_{calc} - (A_i)_{exp}); \quad A_i = (A_i)_{calc} = f \quad [\text{IV.2.5}]$$

Como se puede observar para determinar \mathbf{H} y \mathbf{g} se han de calcular previamente las derivadas de la función con respecto a cada uno de los parámetros que se pretende optimizar. Para llevar a cabo el cálculo de las derivadas numéricas se utiliza el método de diferencias centrales cuya expresión se indica en la ecuación [III.3.48]. El cálculo computacional de las derivadas numéricas correspondientes a la función “ f ” con respecto a las constantes de velocidad se lleva a cabo mediante la subrutina **DERIVk.FOR** escrita en FORTRAN 90. El diagrama de flujo de la Figura IV.2.5 esquematiza el procedimiento computacional que se sigue para el cálculo de la derivada numérica de una función “ f ” con respecto a los parámetros X_1, X_2, \dots, X_p . Una vez calculadas las derivadas numéricas con respecto a cada parámetro a optimizar, el programa **KINMODEL(AGDC)@Vs.1.0** ⁽⁶⁸⁻⁶⁹⁾ determina el valor de las componentes del vector gradiente \mathbf{g} y la matriz Hessiana \mathbf{H} , de acuerdo con las ecuaciones [IV.2.3] y [III.2.4].

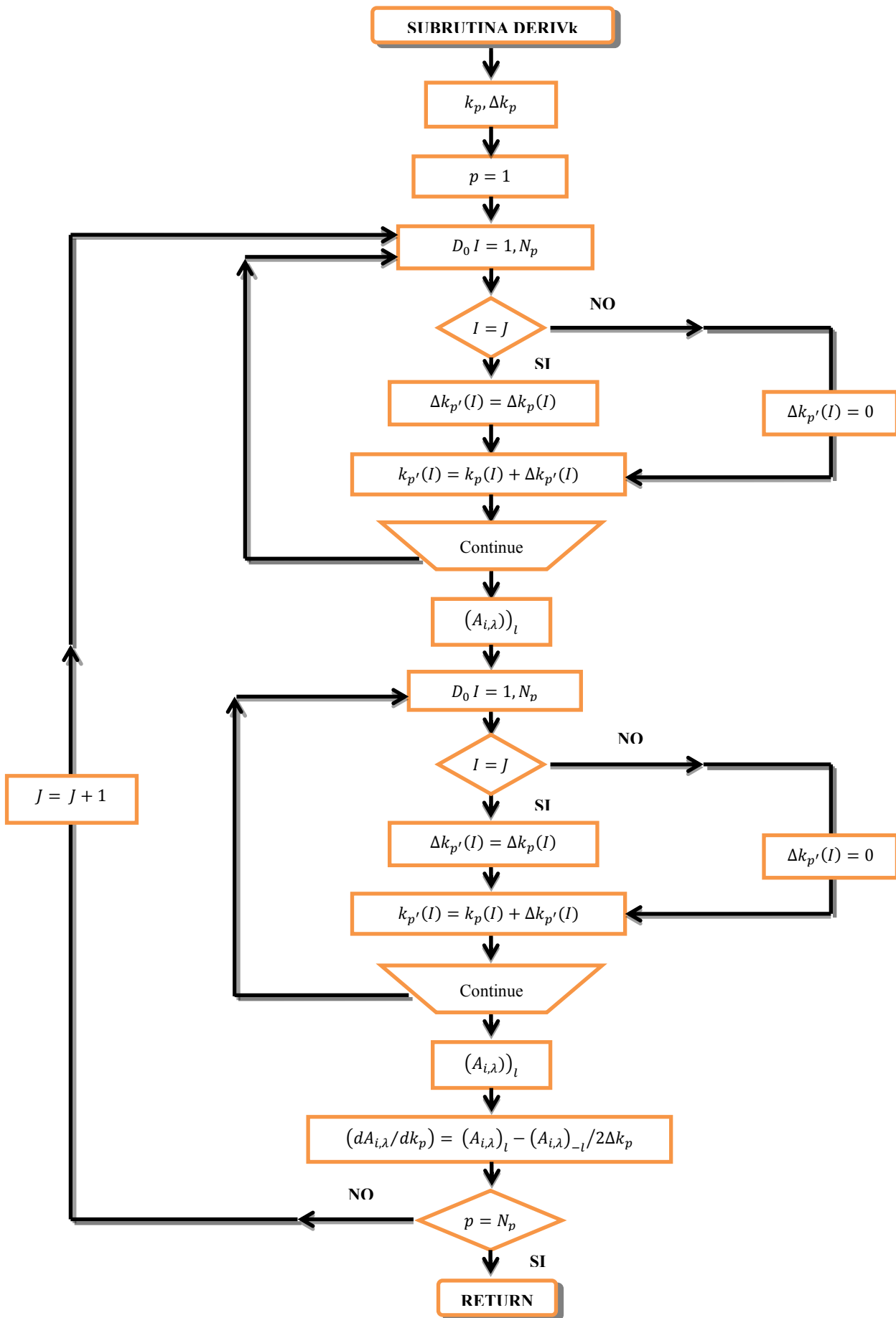


Figura IV.2.5: Diagrama de flujo de la subrutina DERIVk.

IV.2.1.3.- Inversión de la matriz Hessiana

Tal y como hemos visto, para la determinación del vector de movimiento en el algoritmo de optimización AGDC, al tratarse de un método de gradiente de segundo orden, es necesario calcular previamente la inversa de la matriz Hessiana. La determinación de la matriz inversa tiene gran importancia en el proceso de optimización, pudiendo conducir al fracaso del mismo si la inversión no se ha realizado de forma adecuada. Para asegurar la correcta inversión de la matriz Hessiana, se lleva a cabo la misma mediante un proceso en el que se calcula una aproximación inicial de la matriz inversa, mediante el método de eliminación de Gauss. Posteriormente se mejoran los elementos de esta matriz inversa aproximada, mediante un proceso iterativo de aproximaciones sucesivas. Para llevar a cabo el cálculo computacional de la matriz inversa se emplea la subrutina **INVER.FOR**, escrita en **FORTRAN 90**, cuyo diagrama de flujo se representa en la Figura IV.2.7 y que muestra esquemáticamente el proceso seguido para el cálculo de la inversa. A esta subrutina hay que proporcionarle la matriz que se pretende invertir, \mathbf{H} , el número de filas (F_i), el número de columnas, (C_o), y el número de iteraciones deseadas para mejorar el valor de la inversa aproximada.

El procedimiento que se realiza podemos estructurarlo en las siguientes etapas consecutivas:

1. Cálculo del determinante de la matriz a invertir \mathbf{H} , (DetH).
2. Cálculo de la inversa aproximada, \mathbf{H}_m^{-1} . El algoritmo empleado para este cálculo es el de Eliminación de Gauss.
3. Cálculo iterativo de la inversa mejorada, \mathbf{H}_{m+1}^{-1} .
4. Comprobar que se ha alcanzado la exactitud deseada de la inversa: $\mathbf{H}_m^{-1} - \mathbf{H}_{m+1}^{-1} \leq \delta$
5. Comprobar que el producto de la matriz a invertir \mathbf{H} y la inversa obtenida, \mathbf{H}^{-1} , conduce a la matriz unidad \mathbf{I} . $|\mathbf{H}| \mathbf{H}^{-1} = \mathbf{I}$.

El cálculo del determinante (etapa 1) se realiza mediante la subrutina **DETER.FOR**, escrita en **FORTRAN 90**, cuyo diagrama de flujo se muestra en la Figura IV.2.6.

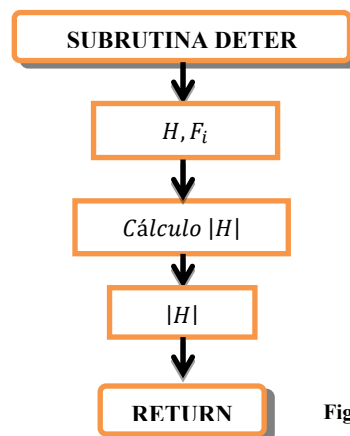


Figura IV.2.6: Diagrama de la subrutina DETER.

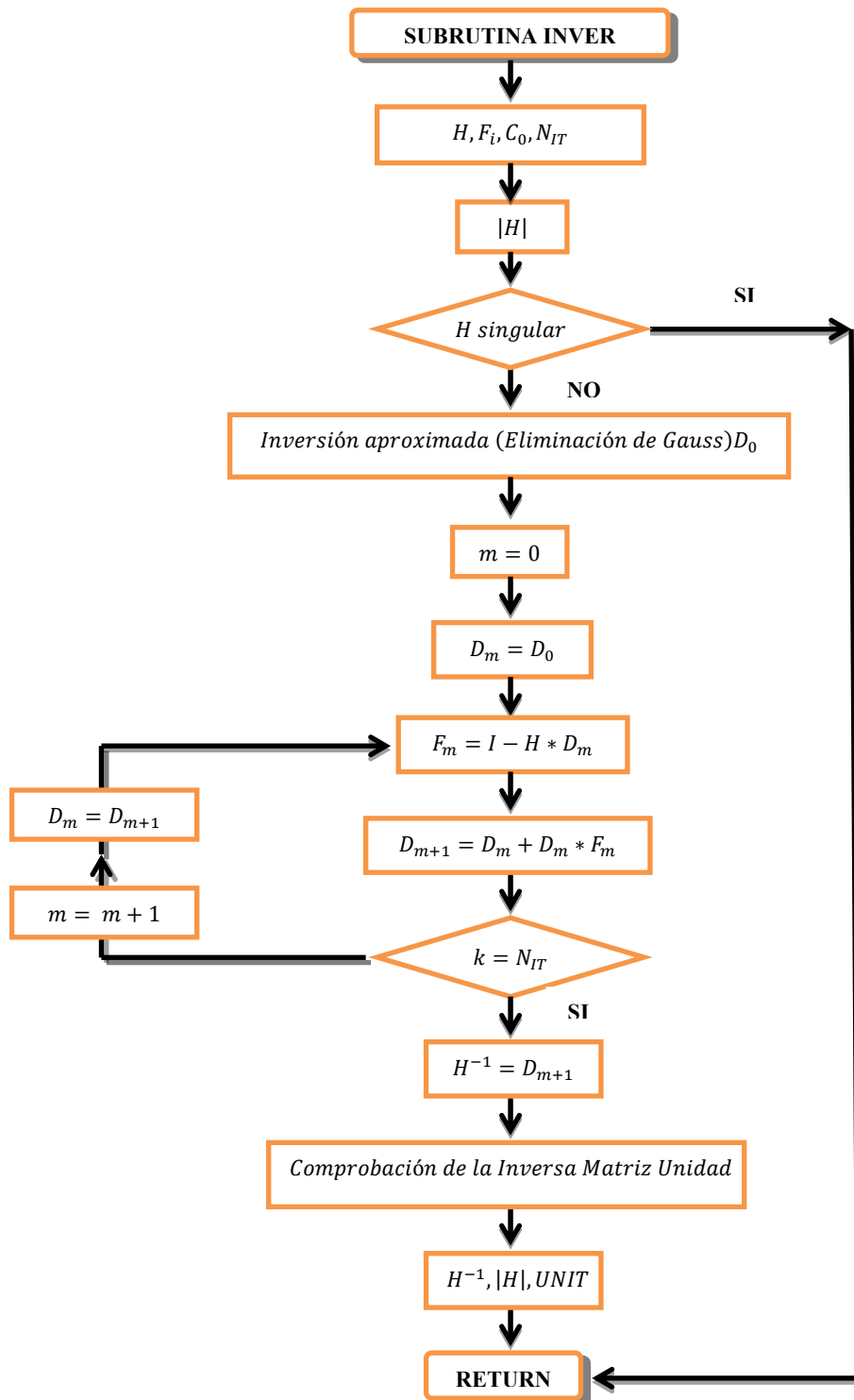


Figura IV.2.7: Diagrama de flujo de la subrutina INVER.

IV.2.1.4.- KINMODEL(AGDC)®Vs.1.0

En los apartados anteriores hemos expuesto el tratamiento computacional de las diferentes partes necesarias para llevar a cabo el proceso de optimización matemática mediante el algoritmo de optimización AGDC. A continuación, vamos a comentar el procedimiento global que se ha seguido para la determinación de las constantes de velocidad. Nuestro objetivo es determinar las constantes individuales de velocidad (k_r) de las distintas etapas del mecanismo de reacción de las N_r reacciones mediante la aplicación del algoritmo de optimización matemática **AGDC**. Para lograrlo se ha utilizado el programa **KINMODEL(AGDC)®Vs.1.0**⁽⁶⁸⁻⁶⁹⁾ que necesita las subrutinas consideradas anteriormente. Mediante la aplicación de este programa, a partir de unos valores estimados de parámetros que pretendemos determinar el avance se produce en la dirección marcada por el vector de movimiento hacia unos valores de los parámetros (k_r) que minimizan el valor de la función SQD . El proceso de optimización matemática responde al diagrama de flujo que se muestra en la Figura IV.2.8.

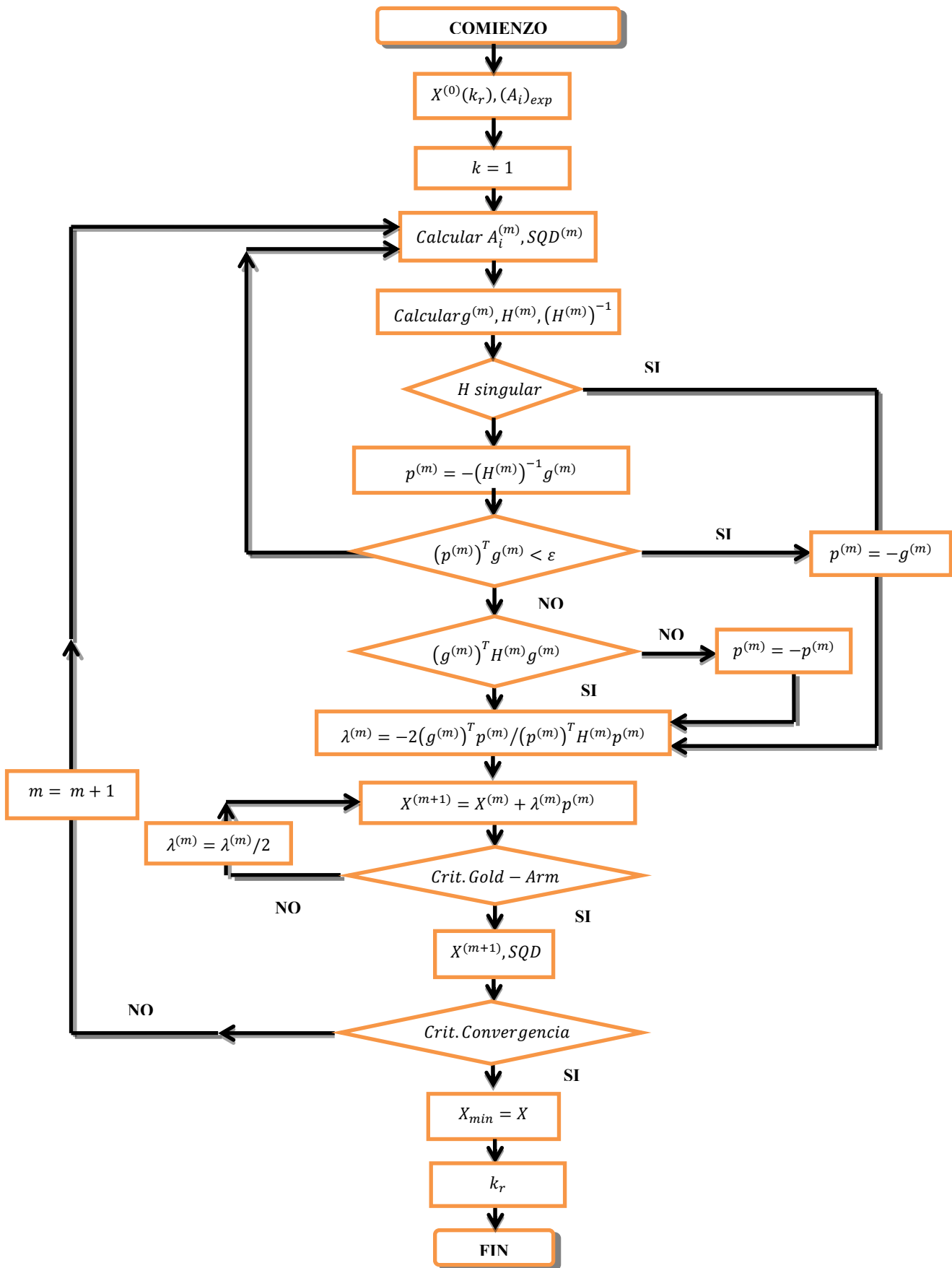


Figura IV.2.8: Diagrama de flujo del proceso de optimización matemática.

El programa **KINMODEL(AGDC)@Vs.1.0** ⁽⁶⁸⁻⁶⁹⁾ se puede representar esquemáticamente de la siguiente manera:

1. Seleccionar el mecanismo de reacción.
2. Seleccionar los parámetros a optimizar (\mathbf{X}).
 - 2.1. Constantes cinéticas (k_r).
 - 2.2. Concentraciones iniciales ($[B_j]_0$).
 - 2.3. Constantes cinéticas y concentraciones iniciales simultáneamente.
 - 2.4. Constantes cinéticas y coeficientes de absorción molar simultáneamente.
3. $m = 0$. Datos de entrada:
 - 3.1. Datos cinéticos experimentales (y_i)_E (absorbancia/tiempo ($(A_{i,\lambda})_{exp}$, concentración/tiempo ($[B_j]$)), criterio de convergencia CC , matriz de coeficientes estequiométricos, $v_{j,r}$, etc...
 - 3.2. Estimaciones iniciales de los parámetros desconocidos $\mathbf{X}^{(0)}$.
4. Determinación de la función $SSQ^{(0)}$.
 - 4.1. Establecimiento y resolución numérica del sistema de ecuaciones diferenciales de velocidad \rightarrow Cálculo de las concentraciones de la especie $[B_j]_i$.
 - 4.2. Cálculo de los valores de absorbancia a cada longitud de onda $(A_{i,\lambda})_{calc}$.

5. ALGORITMO AGDC.

- 5.1. Cálculo del vector de movimiento $\mathbf{p}^{(m)}$.
 - 5.1.1. Calcular las derivadas numéricas de $(A_{i,\lambda})_{calc}$ con respecto a los parámetros a determinar $\mathbf{X}^{(m)}$.
 - 5.1.2. Calcular $\mathbf{g}^{(m)}$ y $\mathbf{H}^{(m)}$
 - 5.1.3. Calcular $(\mathbf{H}^{(m)})^{-1}$
 - 5.1.4. Cálculo de las componentes del vector de movimiento

$$\mathbf{p}^{(m)} = -(\mathbf{H}^{(m)})^{-1} \mathbf{g}^{(m)}$$
- 5.2. Control y corrección de la dirección del vector de movimiento $\mathbf{p}^{(m)}$.
 - 5.2.1. Si $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$, ir a 5.3.
 - 5.2.2. Si $(\mathbf{p}^{(m)})^T \mathbf{g}^{(m)} < \varepsilon$ ($\varepsilon =$ escalar cercano a cero), $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ ir a 5.3.
 - 5.2.3. Si $\mathbf{p}^{(m)} \mathbf{g}^{(m)} > 0$, $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$.
- 5.3. Control de la longitud de $\mathbf{p}^{(m)}$.
 - 5.3.1. Calcular el escalar ($\lambda^{(m)}$) mediante el método de Hartley ⁽⁶⁶⁾.
 - 5.3.2. $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \lambda^{(m)} \mathbf{p}^{(m)}$.
 - 5.3.3. Determinar la función $SQD^{(m+1)}$.

5.3.4. Si se cumple el criterio de Goldstein-Armijo ⁽⁶⁷⁾ ir a 5.4.

5.3.5. $\lambda^{(m)} = \lambda^{(m)}/2$ ir a 5.3.2.

5.4. Calcular $CON = \left| \frac{SQD^{(m+1)} - SQD^{(m)}}{SQD^{(m)}} \right|$

5.5. Si no se alcanza la convergencia ($CON > CC$), $m = m + 1$, ir a 5.1.

6. $X^{(m+1)} = \text{Parámetros optimizados}$.

7. Cálculo de los errores de los parámetros. Análisis estadístico de Residuales.

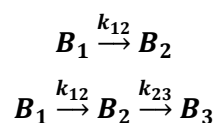
8. Validación de un nuevo modelo e ir a 1.

9. Fin.

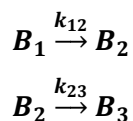
IV.2.2.- Optimización de Parámetros Termodinámicos de Activación mediante el algoritmo AGDC

Los valores de los parámetros de activación determinados previamente mediante la aplicación de la metodología ANN dentro del algoritmo híbrido en todo su contexto, se utilizan como estimaciones iniciales para la mejora de los mismos mediante la aplicación computacional del algoritmo AGDC.

La determinación de los parámetros termodinámicos de activación se lleva a cabo para los siguientes sistemas:



Teniendo en cuenta que la primera etapa del segundo mecanismo es análoga a la única etapa del primer sistema, hemos desarrollado una aplicación computacional que permite la determinación, a conveniencia del usuario, de los parámetros de activación para las siguientes reacciones químicas:



El programa puede realizar los siguientes procesos:

1) Optimización de los parámetros de activación de la reacción: $B_1 \xrightarrow{k_{12}} B_2$

1a.- E_{12} y A_{12}

1b.- ΔS_{12}^\ddagger y ΔH_{12}^\ddagger

2) Optimización de los parámetros de activación de la reacción: $B_2 \xrightarrow{k_{23}} B_3$

2a.- E_{23} y A_{23}

2b.- ΔS_{23}^\ddagger y ΔH_{23}^\ddagger

Tal y como hemos señalado, el proceso de optimización de los parámetros de activación se lleva a cabo mediante el algoritmo de optimización matemática AGDC, en este caso la función SQD a minimizar viene dada por la ecuación:

$$\text{SQD}(\mathbf{X}) = \sum_{i=1}^{N_d} \left((\alpha_j(t_i))_C - (\alpha_j(t_i))_E \right)^2 \quad [\text{IV.2.6}]$$

donde

$$\alpha_j(t_i) = [B_j]_i / [B_1]_0 \quad [\text{IV.2.7}]$$

y \mathbf{X} es el vector que contiene los parámetros a optimizar que pueden ser:

- 1) 1a.- E_{12} y A_{12}
1b.- ΔS_{12}^\neq y ΔH_{12}^\neq
- 2) 2a.- E_{23} y A_{23}
2b.- ΔS_{23}^\neq y ΔH_{23}^\neq

IV.2.2.1.- Cálculo computacional de las derivadas numéricas

La aplicación del algoritmo AGDC implica el cálculo del vector gradiente $\mathbf{g}^{(m)}$ y la matriz Hessiana $\mathbf{H}^{(m)}$ en cada iteración m , para lo cual es necesario calcular las derivadas de $\alpha_j(t_i)$ con respecto a cada uno de los parámetros a optimizar (\mathbf{X}).

Si los residuales vienen dados por:

$$\text{RES}_i = (\alpha_j)_C - (\alpha_j)_E \quad [\text{IV.2.8}]$$

Entonces los elementos de $\mathbf{g}^{(m)}$ y $\mathbf{H}^{(m)}$ son:

$$\mathbf{g} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \text{RES}_i \frac{\partial (\alpha_j)_C}{\partial X_1} \\ \sum_{i=1}^{N_d} \text{RES}_i \frac{\partial (\alpha_j)_C}{\partial X_2} \end{bmatrix}$$

[IV.2.9]

$$\mathbf{H} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_1} \right)^2 & \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_1} \right) \left(\frac{\partial (\alpha_j)_C}{\partial X_2} \right) \\ \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_2} \right) \left(\frac{\partial (\alpha_j)_C}{\partial X_1} \right) & \sum_{i=1}^{N_d} \left(\frac{\partial (\alpha_j)_C}{\partial X_2} \right)^2 \end{bmatrix}$$

[IV.2.10]

El cálculo de las derivadas se lleva a cabo numéricamente mediante el método de diferencias centrales (ecuación [III.3.48])⁽¹³⁸⁾ de acuerdo con el diagrama de flujo de la figura IV.2.9 correspondiente a la subrutina **DERIVX**.

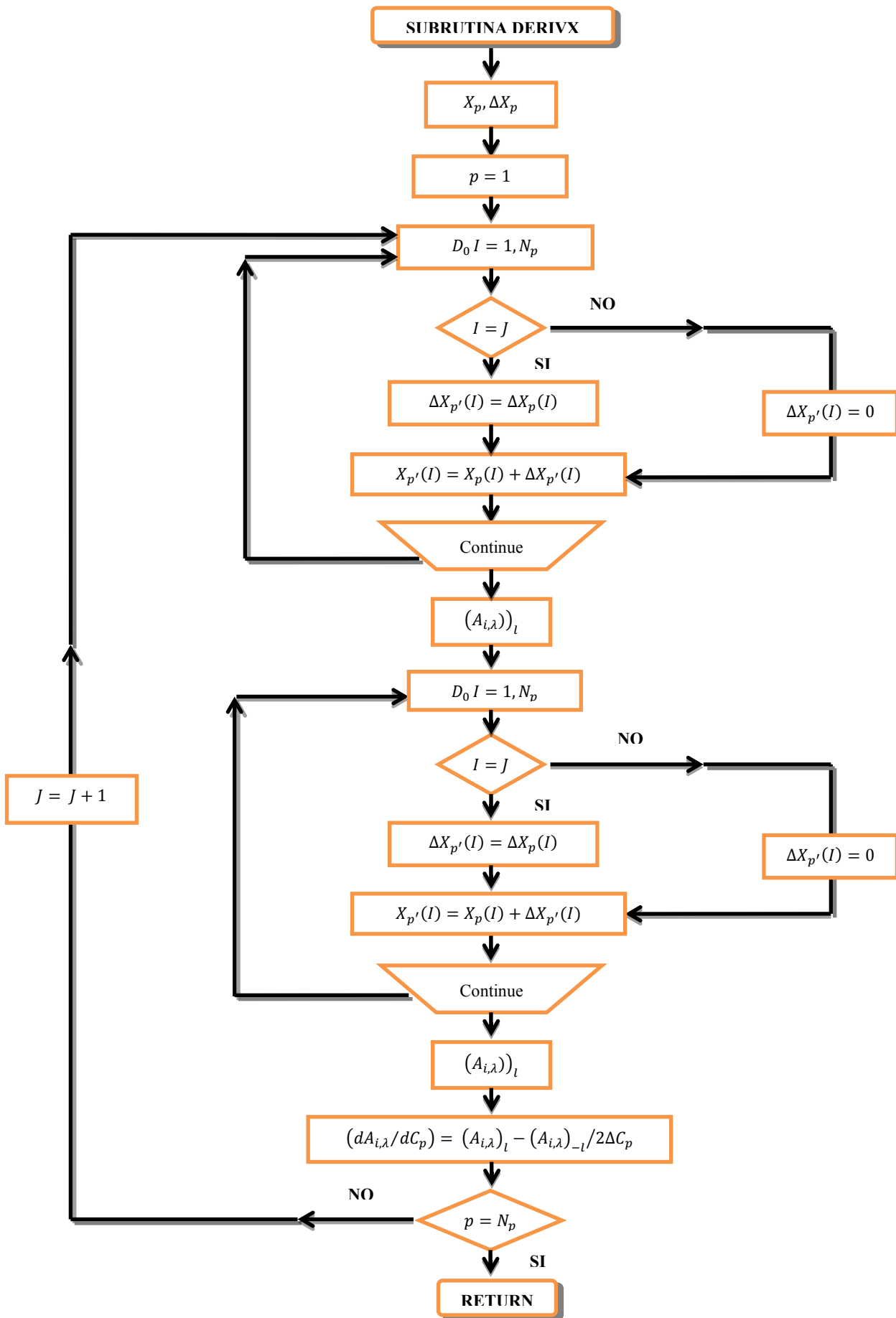


Figura IV.2.9: Diagrama de flujo de la subrutina DERIVX.

IV.2.2.2.- Inversión de la matriz Hessiana

Una vez determinados los términos del vector gradiente y la matriz Hessiana el cálculo del vector de movimiento implica la determinación de la inversa de la matriz Hessiana. Se calcula computacionalmente mediante la *función* MATLAB que computa la inversa de una matriz cuadrada mediante la factorización LU ⁽⁶⁴⁾.

IV.2.2.3.- Algoritmo AGDC

De forma esquemática el algoritmo AGDC se puede escribir de la siguiente manera:

Iteración m → Datos de entrada: $\mathbf{X}^{(m)}$, $SQD^{(m)}$.

1. Cálculo del vector de movimiento.

1.1. Computar las derivadas numéricas parciales de $(\alpha_j)_C$ con respecto a los parámetros a determinar X_p , $(\partial(\alpha_j)_C/\partial X_p)^{(m)}$.

1.2. Computar el vector Gradiente y la matriz Hessiana ($\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$).

1.3. Computar $(\mathbf{H}^{(m)})^{-1}$.

1.4. Calcular los componentes del vector de movimiento $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}[\mathbf{H}^{(m)}]^{-1}$.

2. Control y corrección de la dirección del vector de movimiento $\mathbf{p}^{(m)}$.

2.1. If $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$, ir a 3.

2.2. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} < \varepsilon$ (ε = escalar cercano a cero), $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ ir a 3.

2.3. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} > 0$, $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$.

3. Control de la longitud del vector de movimiento $\mathbf{p}^{(m)}$.

3.1. Computar el escalar ($\lambda^{(m)}$) mediante el método de Hartley ⁽¹³⁴⁾.

3.2. $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \lambda^{(m)}\mathbf{p}^{(m)}$.

3.3. Determinar el valor de la función $SQD^{(m+1)}$ (ecuación [IV.2.6]).

3.4. Si se satisface el criterio de Goldstein-Armijo ⁽¹³⁴⁾ ir a 4.

3.5. $\lambda^{(m)} = \lambda^{(m)}/2$ go to 3.2.

4. Calcular: $CON = \left| \frac{SQD^{(m+1)} - SQD^{(m)}}{SQD^{(m)}} \right|$

5. Si la convergencia no se alcanza ($CON > CC$), entonces $m = m + 1$ ir a 1.

6. $\mathbf{X}^{(m+1)}$ = Parámetros Optimizados.

7. Fin de la Optimización.

El proceso de optimización matemática responde al diagrama de flujo que se muestra en la Figura IV.2.10.

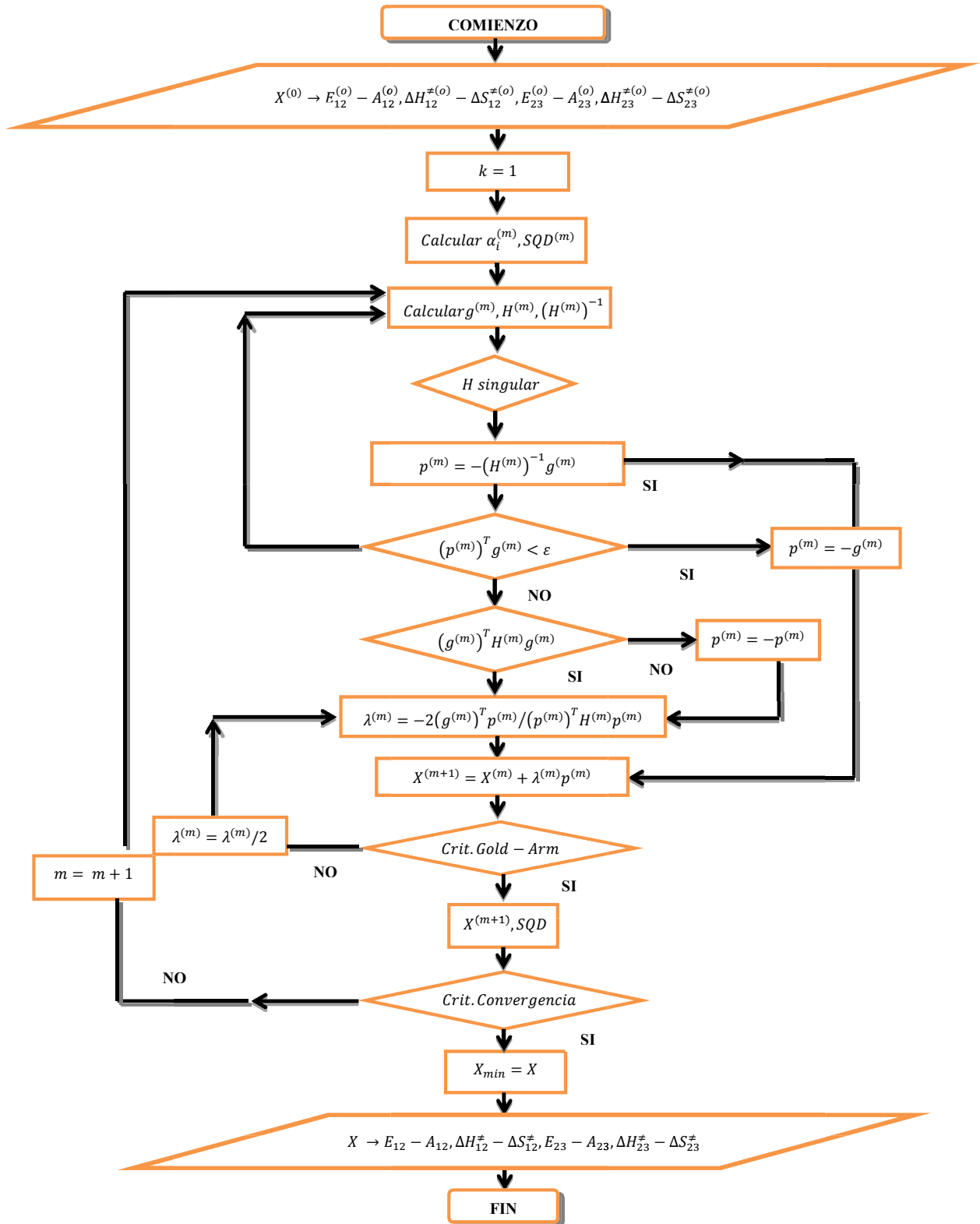


Figura IV.2.10

Durante el proceso de optimización el algoritmo AGDC realiza un riguroso análisis y control del vector de movimiento y de cada uno de sus términos, realizando modificaciones si se detecta algún error, garantizando de este modo el éxito del proceso de optimización. Una vez que ha concluido el proceso de optimización el programa determina los errores de los parámetros ⁽¹⁵⁴⁾ y tras un exhaustivo análisis estadístico de los residuales que permitirá comprobar la *bondad* del ajuste ⁽¹⁵⁵⁾.

IV.2.2.4.- KINNOISOT(AGDC)

La aplicación del algoritmo de optimización matemática AGDC para el tratamiento de cinéticas no isotérmicas se lleva a cabo mediante una aplicación computacional KINNOISOT(AGDC) escrita en el lenguaje propio de MATLAB ⁽⁶⁴⁾. A continuación se expone de forma esquemática el procedimiento computacional que se sigue en la determinación de los parámetros de activación:

0. Datos de entrada: Estimaciones iniciales de los parámetros desconocidos: $\mathbf{X}^{(0)}$ ($E_{12}^{(0)}$, $A_{12}^{(0)}$, $\Delta S_{12}^{\neq(0)}$, $\Delta H_{12}^{\neq(0)}$, $E_{23}^{(0)}$, $A_{23}^{(0)}$, $\Delta S_{23}^{\neq(0)}$, $\Delta H_{23}^{\neq(0)}$). Estos valores corresponden a los de salida (*outputs*) de la aplicación ANN de acuerdo a la estrategia del algoritmo híbrido.

1. Optimización de: E_{12} , A_{12} , ΔS_{12}^{\neq} y ΔH_{12}^{\neq}

$$1a. \mathbf{X} = [E_{12}, A_{12}]. m=0. \mathbf{X}^{(0)} = [E_{12}^{(0)}, A_{12}^{(0)}]$$

1a.1. Calcular $(\alpha_1^{(0)})_C$ [IV.2.7]

1a.1.1) $T = f(t)$ inversa hiperbólica $1/T = (1/T_0) - mt \rightarrow$ Solución matemática exacta $\alpha = e^{-[(|v_1|A_{12}/m E_{12}) (e^{-E_{12}/RT_0})(e^{(m E_{12}/R)t} - 1)]}$

1a.1.2) $T = f(t)$ Polinomial de grado n ($T = \sum_{j=0}^n a_j t^j$)

Procedimientos:

1a.1.2.1) Integración numérica de la ecuación:

$$-(1/|v_1|) \ln \alpha_1 = \int_0^t A_{12} e^{-E_{12}/R} [(1/\sum_{j=0}^n a_j t^j)] dt$$

1a.1.2.2) Solución numérica de la ecuación diferencial de velocidad:

$$d[B_1] = -[B_1]A_{12}e^{-E_{12}/RT} dt$$

1a2. Calcular SQD⁽⁰⁾ [IV.2.6]

1a3. Algoritmo AGDC

1a4. Parámetros Optimizados: $\mathbf{X}^* = [E_{12}, A_{12}]$

Cálculo de los errores de E_{12} , A_{12}

$$1b. \mathbf{X} = [\Delta S_{12}^{\neq}, \Delta H_{12}^{\neq}]. m=0. \mathbf{X}^{(0)} = [\Delta S_{12}^{\neq(0)}, \Delta H_{12}^{\neq(0)}]$$

1b1. Calcular $(\alpha_1^{(0)})_C$ [IV.2.7]: $T = f(t)$ Polinomial de grado n

Procedimientos:

1b.1.1) Integración numérica de la ecuación:

$$-\frac{1}{|v_{1,1}|} \ln \alpha_1 = \int_0^t \left(\frac{k_B}{h}\right) T e^{-\frac{\Delta H_{12}^\ddagger}{RT}} e^{\frac{\Delta S_{12}^\ddagger}{R}} dt$$

1b.1.2) Solución numérica de la ecuación diferencial de velocidad:

$$d[B_1] = -[B_1] \left(\frac{k_B}{h}\right) T e^{-\frac{\Delta H_{12}^\ddagger}{RT}} e^{\frac{\Delta S_{12}^\ddagger}{R}} dt$$

1b2. Calcular SQD⁽⁰⁾ [IV.2.6]

1b3. Algoritmo AGDC

1b4. Parámetros Optimizados: $\mathbf{X}^* = [\Delta S_{12}^\ddagger, \Delta H_{12}^\ddagger]$

Calculo de los errores de $\Delta S_{12}^\ddagger, \Delta H_{12}^\ddagger$.

2. Optimización de: $E_{23}, A_{23}, \Delta S_{23}^\ddagger$ y ΔH_{23}^\ddagger

2a. $\mathbf{X} = [E_{23}, A_{23}]$. $m = 0$. $\mathbf{X}^{(0)} = [E_{23}^{(0)}, A_{23}^{(0)}]$

2a1. Calcular $(\alpha_2(t_i)^{(0)})_C, (\alpha_3(t_i)^{(0)})_C$ [IV.2.7]

2a.1.1) Calcular $T_i = \sum_{j=0}^{j=n} a_j t_i^j$

2a.1.2) Calcular $k_{12}(T_i) = A_{12} e^{-E_{12}/R} \sum_{j=0}^{j=n} a_j t_i^j$

2a.1.3) Calcular $k_{23}(T_i) = A_{23}^{(0)} e^{-E_{23}^{(0)}/R} \sum_{j=0}^{j=n} a_j t_i^j$

2a.1.4) Solución numérica de las ecuaciones diferenciales de velocidad:

$$d[B_2]/dt = k_{12}(T)[B_1] - k_{23}(T)[B_2]$$

$$d[B_3]/dt = k_{23}(T)[B_2]$$

2a2. Calcular SQD⁽⁰⁾ [IV.2.6]

2a3. Algoritmo AGDC

2a4. Parámetros Optimizados: $\mathbf{X}^* = [E_{23}, A_{23}]$

Cálculo de los errores de E_{23}, A_{23}

2b. $\mathbf{X} = [\Delta S_{23}^\ddagger, \Delta H_{23}^\ddagger]$ $m = 0$. $\mathbf{X}^{(0)} = [\Delta S_{23}^{\ddagger(0)}, \Delta H_{23}^{\ddagger(0)}]$

2b1. Calcular $(\alpha_2(t_i)^{(0)})_C, (\alpha_3(t_i)^{(0)})_C$ [IV.2.7]

2b.1.1) Calcular $T_i = \sum_{j=0}^{j=n} a_j t_i^j$

2b.1.2) Calcular $k_{12}(T_i) = (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{12}^\ddagger/R} \sum_{j=0}^{j=n} a_j t_i^j e^{\Delta S_{12}^\ddagger/R}$

2b.1.3) Calcular $k_{23}(T_i) = (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{23}^{\ddagger(0)}/R} \sum_{j=0}^{j=n} a_j t_i^j e^{\Delta S_{23}^{\ddagger(0)}/R}$

2b.1.4) Solución numérica de las ecuaciones diferenciales de velocidad:

$$d[B_2]/dt = k_{12}(T)[B_1] - k_{23}(T)[B_2]$$

$$d[B_3]/dt = k_{23}(T)[B_2]$$

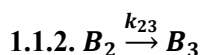
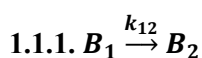
2b2. Calcular SQD⁽⁰⁾ [IV.2.6]

2b3. Algoritmo AGDC**2b4. Parámetros Optimizados:** $\mathbf{X}^* = [\Delta S_{23}^\ddagger, \Delta H_{23}^\ddagger]$ Cálculo de los errores de $\Delta S_{23}^\ddagger, \Delta H_{23}^\ddagger$ **III-** Análisis estadístico de residuales.

El programa KINNOISOT(AGDC) está constituido por un programa principal y una serie de funciones MATLAB (“*subrutinas*”), en las que se realizan los diferentes tratamientos y cálculos necesarios para llevar a cabo el proceso de optimización. Está estructurado en las siguientes partes:

1.- Programa principal → Realiza las siguientes funciones:

1.1. Selecciona la reacción:



1.2. Selecciona los parámetros a optimizar:

1.2.1. $\mathbf{X} = [E_{12}, A_{12}]$

Selecciona el método para calcular $(\alpha_j)_C$

1.2.1.1) $T = f(t) \rightarrow$ Inversa hiperbólica \rightarrow Solución matemática exacta.

1.2.1.2) $T = f(t) \rightarrow$ Polinomial de grado n

Procedimientos:

1.2.1.2.1) Integración numérica.

1.2.1.2.2) Solución numérica de la ecuación diferencial de velocidad.

1.2.2. $\mathbf{X} = [\Delta S_{12}^\ddagger, \Delta H_{12}^\ddagger]$

Selecciona el método para calcular $(\alpha_j)_C$

$T = f(t) \rightarrow$ Polinomial de grado n

1.2.2.1) Integración numérica.

1.2.2.2) Solución numérica de la ecuación diferencial de velocidad

1.2.3. $\mathbf{X} = [E_{23}, A_{23}]$

Selecciona el método para calcular $(\alpha_2(\mathbf{t}_i)^{(0)})_C / (\alpha_3(\mathbf{t}_i)^{(0)})_C$

1.2.3.1) $T = f(t) \rightarrow$ Polinomial de grado $n \rightarrow$ Solución numérica de la ecuación diferencial de velocidad (ODE).

1.2.4. $\mathbf{X} = [\Delta S_{23}^\ddagger, \Delta H_{23}^\ddagger]$

Selecciona el método para calcular $(\alpha_2(\mathbf{t}_i)^{(0)})_C / (\alpha_3(\mathbf{t}_i)^{(0)})_C$

1.2.4.1) $T = f(t) \rightarrow$ Polinomial de grado $n \rightarrow$ Solución numérica de la ecuación diferencial de velocidad (ODE).

1.3. Datos de entrada:

1.3.1. Estimaciones iniciales de los parámetros desconocidos (Valores de salida (*outputs*) de la aplicación ANN):

$$\mathbf{1.3.1.1) X}^{(0)} = [E_{12}^{(0)}, A_{12}^{(0)}]$$

$$\mathbf{1.3.1.2) X}^{(0)} = [\Delta S_{12}^{\neq(0)}, \Delta H_{12}^{\neq(0)}]$$

$$\mathbf{1.3.1.3) X}^{(0)} = [E_{23}^{(0)}, A_{23}^{(0)}]$$

$$\mathbf{1.3.1.4) X}^{(0)} = [\Delta S_{23}^{\neq(0)}, \Delta H_{23}^{\neq(0)}]$$

1.3.2. Datos experimentales de $(\alpha_j)_E/t_i$ ($(\alpha_1)_E/t_i$, $(\alpha_2)_E/t_i$, $(\alpha_3)_E/t_i$).

1.3.3. Concentraciones iniciales $[\mathbf{B}_j]_0$ ($[\mathbf{B}_1]_0$, $[\mathbf{B}_2]_0$, $[\mathbf{B}_3]_0$).

1.3.4. Parámetros cuyos valores son conocidos.

1.3.5. Criterio de Convergencia CC,..

1.4. Datos de salida:

1.4.1. Parámetros Optimizados:

$$\mathbf{1.4.1.1) } E_{12}, A_{12}$$

$$\mathbf{1.4.1.2) } \Delta S_{12}^{\neq}, \Delta H_{12}^{\neq}$$

$$\mathbf{1.4.1.3) } E_{23}, A_{23}$$

$$\mathbf{1.4.1.4) } \Delta S_{23}^{\neq}, \Delta H_{23}^{\neq}$$

1.4.2. Errores de los parámetros.

1.4.3. Análisis estadístico de residuales.

2. Funciones:

2.1. fAGDC, función que lleva a cabo el proceso de optimización mediante la aplicación del algoritmo AGDC.

2.2. fEXACT, función que calcula $[\mathbf{B}_1]_{it}$ y $(\alpha_i)_C$ a partir de la solución matemática exacta.

2.3. fNUMINT1, función que genera e integra numéricamente, mediante los métodos proporcionados por MATLAB: *quad*, *quadl*, *quadv* o *quadgk* ⁽⁶⁴⁾, la ecuación [III.4.34].

2.4. fNUMINT2, función que genera e integra numéricamente, mediante los métodos proporcionados por MATLAB: *quad*, *quadl*, *quadv* o *quadgk* ⁽⁶⁴⁾, la ecuación [III.4.39].

2.5. fDIFEQ, función que establece el sistema de ecuaciones diferenciales de velocidad.

2.6. fNUMSOL, función que realiza la resolución numérica del sistema de ecuaciones diferenciales de velocidad mediante los métodos proporcionados por MATLAB: *ode45*, *ode23*, *ode113* ó *ode15s* ⁽⁶⁴⁾.

2.7. fDERIV, calcula las derivadas numéricas parciales de $(\alpha_j)_C$ con respecto a los parámetros que van a ser optimizados mediante el método numérico de diferencias centrales.

2.8. finVER, calcula el determinante y realiza la inversión de la matriz Hessiana mediante la función *inv*⁽⁶⁴⁾ proporcionada por MATLAB.

2.9. fESTAD, función que determina los errores de los parámetros optimizados y realiza el análisis estadístico de residuales.

*Huir de la lucha es lo peor que puede sucedernos.
Es peor que perder la lucha, porque en la derrota siempre
podemos aprender algo, pero en la fuga todo lo que logramos
es declarar la victoria de nuestro Enemigo.
El peregrino de Compostela (Diario de un mago)*

V. Diseño de Experimentos y Organización matricial de datos numéricos

Capítulo V.

Diseño de Experimentos y Organización matricial de datos numéricos.

V.1.- Diseño de Experimentos (ED)

El término *Diseño de Experimentos* (ED) se usa habitualmente en 2 contextos diferentes. El primero de ellos describe el conjunto de experiencias que hay que realizar para desarrollar o probar un modelo idóneo que represente fehacientemente un determinado proceso observado. El segundo se usa con la intención de optimizar procesos buscando las condiciones más favorables de obtención de productos que puedan imponerse a dichos procesos con el fin de obtener el máximo rendimiento. Este es precisamente el propósito que pretendemos alcanzar en el presente trabajo con la utilización de un adecuado ED. Por tanto, se trata de planificar un conjunto de experimentos que contenga la más amplia información relativa al sistema o evento que se va a estudiar.

Se consideran modelos estadísticos clásicos aquellos cuyo objetivo es averiguar si unos determinados agentes, denominados *Factores*, influyen de manera cualitativa en alguna variable del sistema o proceso en estudio. Si se detecta la influencia de algún *Factor*, es necesario cuantificar dicha influencia. Por tanto, en el *Diseño de Experimentos* se consideran 2 tipos de variables, los *Factores* susceptibles de poseer algún tipo de influencia sobre el sistema y las *Respuestas* a la variación de estos agentes. Se denominan *Factores* a las variables que se modifican de manera controlada para estudiar su efecto cuantitativo en el sistema o proceso objeto de estudio y se consideran como *Respuestas* las propiedades del sistema o las características del proceso. En nuestro caso los *Factores* son las constantes de velocidad (k_r) cuando trabajamos con cinéticas isotérmicas y los Parámetros Termodinámicos de Activación (PTA) en el estudio de cinéticas no isotérmicas. Las *Respuestas* son: a) las variables absorbancia total de la mezcla, b) las absorbancias y concentraciones individuales de cada especie química involucrada en el sistema reaccionante para cinéticas isotérmicas y c) la fracción remanente individual para cada especie en el caso de cinéticas no isotérmicas. Las *Respuestas* son las variables dependientes (Y_j) y los *Factores* son las variables independientes (X_j), que podemos representar de acuerdo a la ecuación:

$$(Y_1, Y_2, \dots, Y_n) = f(X_1, X_2, \dots, X_n) \quad [\text{V.1.1}]$$

En muchas ocasiones las *Respuestas* se consideran individualmente como ocurre en nuestro caso. Por tanto:

$$(Y) = f(X_1, X_2, \dots, X_n) \quad [\text{V.1.2}]$$

Los modelos se obtienen de las experiencias. El término “Diseño” significa que las experiencias no se realizan de una manera aleatoria sino de una forma controlada, sistematizada y planificada “*a priori*”. El ED se usa para poder disponer de un proceso, como es la cinética de una reacción química con las características deseadas. Es decir se pretende:

a) averiguar el efecto de los *Factores*.

b) evaluar la relación X/Y , es decir *Factor/Respuesta*, con un mínimo, pero significativo y suficiente número de experiencias sistematizadas.

La influencia de los *Factores* sobre las *Respuestas* puede ser de tipo cualitativo o cuantitativo. Así, la influencia de varios catalizadores sobre la velocidad de una reacción es de tipo cualitativo y la influencia del pH es de tipo cuantitativo. En nuestro caso, la influencia de las constantes de velocidad sobre la/s absorbancia/s o concentraciones es de tipo cuantitativo, del mismo modo que la influencia de los PTA sobre la/s fracciones remanentes, cuya magnitud tenemos que evaluar.

V.1.1.- Diseño Experimental Factorial Completo (CFED)

Para realizar un Diseño de Experimentos el primer paso consiste en elegir los *Factores* y los extremos del *Dominio Experimental* (DE) que se denominan *Niveles*. Un *Diseño Experimental Factorial Completo* (*Complete Factorial Experimental Design* (CFED)) es aquel que consta de dos o más *Factores*, cada uno de los cuales con distintos valores de *Niveles* y cuyas unidades experimentales cubren todas las posibles combinaciones de esos *Niveles* para todos los *Factores*. Este tipo de experimentos permite el estudio del efecto individual de cada *Factor* sobre la variable *Respuesta*, así como el efecto de las interacciones entre *Factores* sobre dicha variable. El número de experimentos (n) de un diseño CFED se incrementa de manera exponencial al aumentar el número de *Factores* (k) y el de *Niveles* (l), puesto que el número de experimentos (n) a realizar para un sistema con k *Factores* y 2 *Niveles* ($l=2$) viene dado por la fórmula:

$$n = 2^k \quad [\text{V.1.3}]$$

El caso más sencillo de un Diseño de Experimentos es aquel con un solo *Factor* y 2 *Niveles*. Para la evaluación de las *Respuestas* en este caso tendremos que realizar solo 2 experimentos y el *Dominio Experimental* se configura en un espacio monodimensional (1D). Si consideramos 2 *Factores* con 2 *Niveles*, tendremos que realizar 4 experimentos ($n=4$) configurando su *Dominio Experimental* en un espacio bidimensional (2D), determinado por un cuadrilátero con formas simétricas (cuadrangular -Figura V.1.1- o rectangular), o bien no simétricas (trapezoidal asimétrica). Si consideramos 3 *Factores* con 2 *Niveles*, el número de experimentos será 8 ($n=8$) configurando su *Dominio Experimental* en un espacio tridimensional

(3D) definiendo un espacio poliédrico en forma de hexaedro regular cuadrangular o de prisma rectangular (Figura V.1.2).

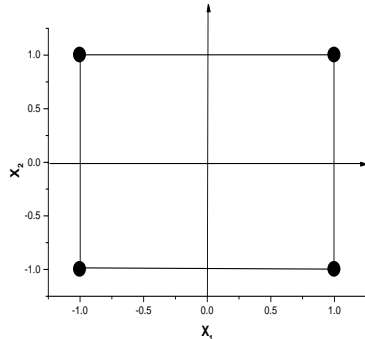


Figura V.1.1: CFED en un espacio bidimensional.

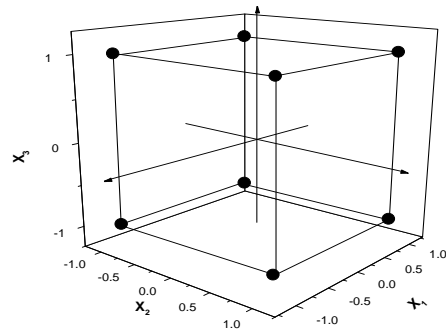


Figura V.1.2: CFED en un espacio tridimensional.

Esto es para la variación simultánea de los 3 *Factores*, sin embargo hay que tener en cuenta que cuando existen más de 2 *Factores*, tendremos muchos más efectos al considerar variaciones conjuntas binarias, ternarias, etc. Así, si consideramos 7 *Factores* con 2 *Niveles*, tendremos 128 experimentos (2^7), es decir 7 efectos principales de 1 *Factor*, 21 de 2 *Factores*, 35 de 3 *Factores*, 35 de 4 *Factores*, 21 de 5 *Factores*, 7 de 6 *Factores* y 1 de 7 *Factores*. La experiencia indica que los experimentos con más de 3 *Factores* resultan irrelevantes. Por esto el número de combinaciones en un *Diseño Experimental Factorial Completo* (CFED), es demasiado alto para su procesamiento y puede optarse por un *Diseño de Experimentos Factorial Fraccional* (FFED), en el que se omiten algunas de las combinaciones posibles. La notación habitual del Diseño FFED es:

$$n = 2^{(k-m)} \quad \text{[V.1.4]}$$

donde m es el número de experimentos cuya realización se propone no realizar por considerarlos irrelevantes y carentes de eficacia.

Si aumentamos el número de *Niveles* aumenta enormemente el número de experimentos (n) del diseño CFED, pues la fórmula de obtención es:

$$n = 3^k \quad \text{[V.1.5]}$$

Así por ejemplo, en el diseño CFED para un sistema con 2 *Factores* y 3 *Niveles* hay que realizar 9 experimentos, (Figura V.1.3), mientras que para 3 *Factores* y 3 *Niveles* tendremos 27 experimentos y 81 para 4 *Factores* y 3 *Niveles*.

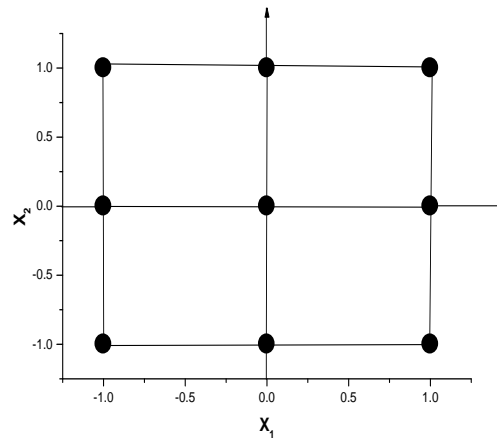


Figura V.1.3: CFED para un sistema con 2 Factores y 3 Niveles en el que se han realizado 9 experimentos.

Si aumentamos simultáneamente el número de *Factores* y *Niveles*, aumentan la base y el exponente de la potencia de la fórmula [V.1.3] y el colectivo de experimentos a realizar es extraordinariamente elevado, lo cual hace prácticamente imposible el poder llevarlo a cabo en la práctica. Para resolver este problema se acude al denominado *Diseño de Experimentos Central Compuesto* (*Central Composite Experimental Design* (CCED))⁽¹⁵⁶⁾ que describimos a continuación.

V.1.2.- Diseño de Experimentos Central Compuesto (CCED)

Consiste en realizar una composición de diseños añadiendo a uno CFED básico, otros diseños como es el que se organiza en “estrella”, denominado *Star Experimental Design* (SED), dando lugar al diseño *Star Composite Experimental Design* (SCED) (Figuras V.1.4 y V.1.5) que es muy útil en el campo Computacional y que nos ha servido de pauta para generar las experiencias sintéticas del presente trabajo en la mayoría de los casos. De esta manera, si a los experimentos considerados en un *Diseño de Experimentos Factorial Completo* (CFED) de 2 Factores (X_1, X_2) y 2 Niveles que nos proporciona 4 experimentos (Experiencias 1 a 4 de la Tabla V.1.1, señaladas con la letra A), le añadimos los Experiencias 5 a 8 (Tabla V.1.1, señaladas con la letra B) que se ubican en forma de “estrella” (SCED) con 2 Niveles y separaciones $\pm\alpha = \sqrt{2}$ y por último añadimos la Experiencia número 9 (Tabla V.1.1), cuyo nodo se ubica en el origen de coordenadas (señalado con la letra C), tendremos la composición (ABC) denominada *Central Star Composite Experimental Design* (CSCED). (Figura V.1.4).

Exper.	X ₁	X ₂	
1	-1	-1	(A)
2	+1	-1	(A)
3	-1	+1	(A)
4	+1	+1	(A)
5	$-\sqrt{2}$	0	(B)
6	$+\sqrt{2}$	0	(B)
7	0	$-\sqrt{2}$	(B)
8	0	$+\sqrt{2}$	(B)
9	0	0	(C)

Tabla V.1.1: Composición para un diseño del tipo CSCED (*Central Star Composite Experimental Design*).

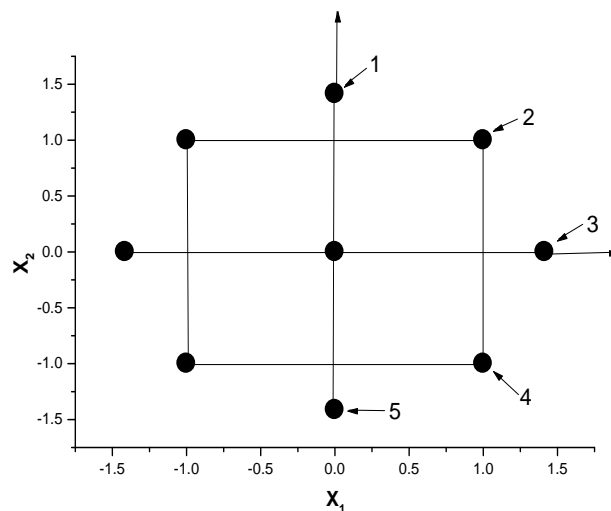


Figura V.1.4: Composición denominada *Central Star Composite Experimental Design* (CSCED).

El diseño CSCED tiene indudables ventajas respecto a los diseños *Sencillos* (*No Compuestos*) del tipo CFED. Así, comparando el CSCED de la Figura V.1.4 con el CFED de la Figura V.1.3 que posee 2 *Factores* y 3 *Niveles*, observamos que ambos poseen el mismo número de experimentos ($n=9$) sin embargo en el CFED cada *Factor* se encuentra en 3 niveles (Figura V.1.3) mientras que en el CSCED se encuentra en 5 *Niveles* (puntos señalados con los números 1 al 5 en la Figura V.1.4) consiguiendo una configuración con una distribución mucho más efectiva puesto que, con el mismo número de experimentos, se abarca una mayor “superficie experimental” (*Dominio Experimental*). A medida que aumenta el número de *Factores* y *Niveles*, se incrementan las ventajas. Así por ejemplo, al comparar las configuraciones del *Diseño Simple* CFED correspondiente a 3 *Factores* y 3 *Niveles* con la del *Diseño Compuesto* (CSCED) (Figura V.1.5), observamos 2 efectos a) el número de experimentos a realizar se reduce desde los 27 (3^3) experimentos en el diseño CFED a

únicamente 15 en el CSCED y **b)** cada *Factor* se encuentra en 5 *Niveles* (CSCED) en lugar de los 3 *Niveles* del CFED, abarcando un mayor volumen o *Dominio Experimental* con muchos menos experimentos a realizar. Es decir, la conclusión es que el *Diseño Compuesto* (CSCED) implica un importante ahorro de trabajo experimental y consigue unas configuraciones mucho más efectivas que los *Diseños Sencillos* del tipo CFED. La Figura V.1.5 muestra un caso real de aplicación del diseño CSCED con 3 *Factores* y 2 *Niveles*, correspondiente al sistema cinético de reacciones consecutivas, en el que la primera reacción es reversible y la segunda irreversible y en el que se involucran 3 especies (B_1 , B_2 y B_3) con 3 *Factores* (las 3 constantes cinéticas k_{12} , k_{21} y k_{23}) y 2 *Niveles*.

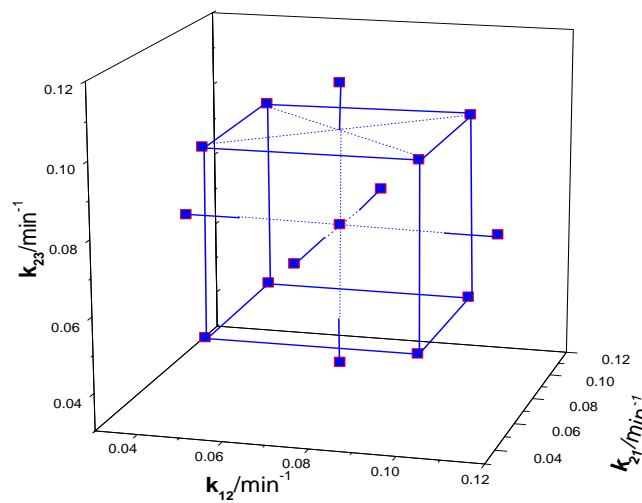


Figura V.1.5: CSCED con 3 *Factores* y 2 *Niveles* para el sistema cinético de reacciones consecutivas.

Hemos comentado anteriormente que en el *Central Star Composite Experimental Design*, (CSCED), los puntos extremos de la “estrella” se encuentran a una distancia $\pm\alpha = \sqrt{2}$. Pues bien, esta distancia no obedece a una elección aleatoria sino que corresponde al origen de coordenadas que hace que los 8 puntos exteriores sean equidistantes, es decir, los correspondientes extremos de la estrella y los de los vértices del prisma hexaédrico regular (cubo) pertenezcan a una circunferencia de diámetro igual a la diagonal del prisma ⁽¹⁸⁾. En otras ocasiones $\pm\alpha = 1$, configurando un *Diseño Experimental de Cubo Centrado en las Caras* y cuando razones de tipo experimental imposibilitan asignar un valor de $\alpha \geq 1$ se elige un valor menor de $\pm\alpha = 1/\sqrt{2}$ (puesto que $\alpha = \sin 45^\circ = 1/\sqrt{2}$) que da lugar a un *Diseño Experimental de Cubo Inscrito*.

El procedimiento de Diseño de Experimentos CSCED admite además la adición de otros experimentos del tipo *Factorial Completo* (CFED), dando lugar a nuevas y más eficaces

configuraciones. Se pueden añadir otras ordenaciones distintas de las básicas de 2 *Factores* y 3 *Niveles*, correspondientes a nuevas variaciones conjuntas de ambos *Factores* y que son distintas a las de los valores extremos del *Nivel* anterior. De esta manera se originan más configuraciones, aumentando la secuencia de cuadrados inscritos y circunscritos al que se ha tomado como base (Figura V.1.6), resultando un diseño más completo y eficaz. Con ello, se abarca un mayor *Dominio Experimental* ya sea en *Superficie* o *Volumen Experimental*, que nos va a permitir tener una información experimental más amplia que supone una mejora de los procesos de *Aprendizaje* y *Entrenamiento* de las Redes Neuronales. Las evidentes ventajas que ofrece el tipo de *Diseños de Experimentos Compuestos* (EDCSC), han aconsejado que sea éste el que hayamos adoptado con la totalidad de nuestro trabajo. Un caso particular se muestra en la Figura V.1.6 en la que hay 2 *Factores* (k_1 y k_2) correspondientes a las constantes cinéticas de uno de los sistemas estudiados, (modelo de reacciones consecutivas) en las que se involucran 3 especies (B_1 , B_2 y B_3).

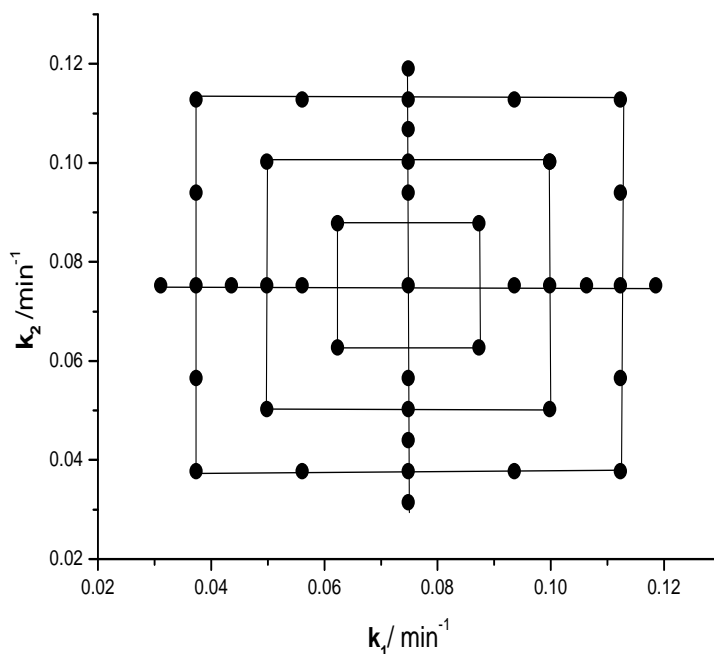


Figura V.1.6: EDCSC con 2 *Factores* y se involucran las 3 especies del modelo de reacciones consecutivas.

Es necesario realizar una consideración final, ya que nuestro ED no persigue la organización de experimentos que se van a realizar “*a posteriori*” en el laboratorio y donde los *Factores* son las variables experimentales que se pueden modificar a voluntad para evaluar su influencia en la/s *Respuesta/s*. En nuestro caso las variables son las constantes cinéticas (k_r) y/o los Parámetros Termodinámicos de Activación (PTA) correspondientes a cada reacción elemental. Sin embargo, el procedimiento que hay que utilizar, es idéntico y resulta idóneo para nuestros objetivos de *Aprendizaje* o *Entrenamiento* de las Redes Neuronales. Tenemos que

generar *a priori* un conjunto de experiencias sintéticas, organizadas de acuerdo al *Diseño de Experimentos* elegido y que hemos considerado óptimo (*Central Star Composite Experimental Design*) (CSCED), cuyos nodos los hemos obtenido a partir de las variaciones controladas de las variables que en nuestro caso son precisamente las constantes cinéticas (k_i) y los Parámetros Termodinámicos de Activación (PTA).

V.2.- Organización matricial de curvas y datos cinéticos: Computación ANN y Representación gráfica (“visualización”)

Debido a la enorme cantidad de curvas y datos numéricos de los sistemas cinéticos que hay que manejar para el tratamiento computacional con ANN, es necesario ordenarlos y sistematizarlos convenientemente para conseguir un doble propósito:

a) satisfacer las exigencias relativas a las propiedades, dimensiones y demás requisitos exigidos por las aplicaciones de software informático para el tratamiento computacional ANN (MATLAB⁽⁶⁴⁾ y TRAJAN⁽⁶¹⁾) principalmente con las matrices de entrada (*Inputs*) garantes del éxito del tratamiento.

b) realizar una representación gráfica adecuada (generalmente 3D) que sea lo suficientemente ilustrativa, en orden a que sirva de ayuda para la visualización e interpretación de las curvas y datos en su conjunto. Obviamente, deben satisfacer los requisitos exigidos por las herramientas y aplicaciones del software de trazado de figuras (MATLAB, ORIGIN, EXCEL, QTILOT,...etc.).

Dichas herramientas de software que permiten trazar las figuras de las representaciones gráficas de grandes colectivos de datos, poseen un amplio abanico de posibilidades que acercan la realidad tridimensional a la ficción bidimensional que el ojo humano es capaz de visualizar. Parece pues oportuno ensayar varias alternativas y elegir la más conveniente. La posibilidad de movimiento de la figura 3D que permite la elección de la perspectiva más adecuada, debe ser igualmente ensayada en orden a visualizar más fácilmente la figura para que resulte más ilustrativa. Además, se debe realizar la representación conjunta de varios grupos de datos (por ejemplo los pertenecientes a series de sistemas cinéticos comunes) sin que haya una pérdida en la claridad de la figura, en orden a visualizar concordancias o discordancias de las tendencias y comportamientos entre los distintos grupos. Asimismo, es de gran ayuda la inclusión de los denominados *Mapas de Contorno* como resultado de la proyección de figuras 3D en superficies bidimensionales, sobre el plano base de la representación gráfica. La organización de datos de las distintas curvas cinéticas que conforman las matrices de entrada (*Inputs*) y que van a someterse a tratamiento computacional por las distintas aplicaciones de software ANN, ha de satisfacer los requisitos que exigen dichas aplicaciones. Esto es, las matrices de entrada *Inputs* han de ser compatibles con las dimensiones de las matrices de parámetros “objetivos” (*Targets*),

para poder ejecutar de manera adecuada los procesos de *Entrenamiento* o *Aprendizaje* y *Predicción* de la Red neuronal en estudio.

V.2.1.- Tipos de datos cinéticos

Una curva cinética tradicional corresponde al perfil de variación con el tiempo de la concentración o preferiblemente, de una magnitud químico física relacionada con ella. En nuestro caso, cuando se determinan las constantes individuales de un mecanismo, la magnitud químico física cuya evolución se controla es la absorbancia monitorizada con el tiempo, cuyos puntos van a constituir el conjunto de datos numéricos objeto de computación por ANN. Estas curvas pueden obtenerse en muy diversas condiciones ya que dependen de muchas variables experimentales, como son la longitud de onda, las concentraciones iniciales de las especies, temperatura, fuerza iónica...etc. Hay que aprovechar esta *multidependencia* para obtener un colectivo de datos cinéticos y un número de curvas lo más numeroso posible, ya que estas curvas y datos, contienen toda la información del sistema cinético objeto de estudio. Es obvio que cuanto más amplio sea el colectivo, más elevado será el número de parámetros cinéticos que podremos determinar y más alto su grado de exactitud y precisión. En el caso de la determinación de Parámetros Termodinámicos de activación (PTA) se obtiene la variación continua de la concentración con el tiempo para posteriormente obtener la fracción remanente α .

En el tratamiento ANN se pueden computar datos cinéticos con distintas características y procedencias que constituyen las matrices *Inputs*. Las matrices que se preparan para representación gráfica y visualización han de estar de acuerdo con diferentes criterios de ordenación. Pueden ser de los siguientes tipos:

- a) valores de la absorbancia total de la mezcla reaccionante adquiridos a una determinada longitud de onda (SW).
- b) valores de las absorbancias individuales, correspondientes a cada especie por separado a determinada/s longitud/es de onda (SW/MW) y que han sido calculados (no medidos experimentalmente) a partir de los correspondientes valores de los coeficientes de absorción molar (ϵ_j^{λ}) y/o valores de las fracciones molares remanentes individuales para cada una de las especies.
- c) valores de las concentraciones individuales de cada especie calculadas a partir de los valores de las absorbancias individuales.
- d) valores de las fracciones remanentes individuales de cada especie calculadas a partir de las concentraciones individuales.

V.2.2.- Elección del tipo idóneo de datos cinéticos isotérmicos y no isotérmicos

Cuando evaluamos con cinéticas isotérmicas, los valores de las absorbancias totales de la mezcla reaccionante son los únicos que pueden medirse experimentalmente en el laboratorio. En determinadas ocasiones no es posible realizar la computación ANN de este tipo de datos (posteriormente analizaremos los motivos) y hay que utilizar los n_j datos correspondientes a las absorbancias individuales de cada especie o incluso los n_j datos de las concentraciones. Sin embargo, estos valores generalmente no pueden adquirirse mediante medidas experimentales directas y para acceder a su conocimiento, hay que realizar (n_j-1) medidas adicionales de absorbancia total, al mismo valor de tiempo y a longitudes de onda distintas. Hay que tener en cuenta que (n_j-1) es el número de concentraciones que son linealmente independientes, ya que existe una relación entre ellas (la ecuación del balance de materia) que consume una libertad. De esta manera, se plantea un sistema de ecuaciones, constituido por dicha relación de balance de materia y (n_j-1) funciones de los valores de la absorbancia total de la mezcla expresadas como suma de los productos cruzados de los $\varepsilon_j^{\lambda_j}$ y las concentraciones de todas y cada una de las especies. Por tanto, tenemos un sistema de n_j ecuaciones con n_j incógnitas y cuyas soluciones, conociendo *a priori* los valores de $\varepsilon_j^{\lambda_j}$, nos van a permitir conocer las n_j absorbancias individuales $(A_j^{\lambda_j})$ o las n_j concentraciones individuales $([B_j])$. Se calculan a partir de las medidas de absorbancia total de la mezcla $(A_{T,t_i,n_c}^{\lambda_j})$ realizadas al tiempo t_i , a la longitud de onda λ_j en la curva correspondiente a la reacción n_c , con una longitud de paso óptico de 1 cm.

Esta situación es muy frecuente y ocurre cuando en el espectro global UV-Vis de la mezcla reaccionante, obtenido en un amplio margen de longitudes de onda, hay solapamientos de bandas de todas las especies que presentan absorbancia a la longitud de onda considerada. Ahora bien, el espectro o determinadas zonas del mismo, pueden ofrecer otras posibilidades dentro de una elevada casuística que puede presentarse. Así, pueden detectarse bandas de absorción selectivas para una especie al no existir interferencias con las de otra sustancia. En este caso podríamos conocer directamente y sin necesidad de realizar ningún cálculo posterior, el valor de la absorbancia individual (o concentración) de esa especie ya que coincide con el valor medido de la absorbancia total. Las absorbancias (o concentraciones) individuales de las demás especies pueden calcularse resolviendo los sistemas de ecuaciones aludidos en el Capítulo III.

Otro caso de interés es aquel en el que existen 2 zonas, una en la que hay una banda selectiva y otra en la que el solapamiento tiene lugar entre bandas de 2 o más especies. Para la elección del tipo de datos que van a ser computados por ANN, tenemos que considerar también los factores estrictamente cinéticos ya que tienen una influencia directa en los valores de absorbancias del espectro de la mezcla reaccionante y en la evolución con el tiempo. Se trata de

los valores relativos de las constantes cinéticas, que influyen directamente sobre la evolución de los valores de concentraciones y absorbancias. La magnitud de esta relación determinará los perfiles de las concentraciones y absorbancias así como su variación con el tiempo.

En suma, el procedimiento consiste en el análisis previo del espectro y su evolución con el tiempo tras considerar los 2 tipos de factores (espectroscópicos y cinéticos) cuya influencia hemos evaluado. Esto nos va a permitir adoptar un criterio para obtener los datos cinéticos que resulten más convenientes (absorbancia total, absorbancia individual o concentración) para realizar el tratamiento ANN posterior. Este es el procedimiento que hemos utilizado en el *Diseño de Experimentos* del tipo *Central Star Composite Experimental Design* (CSCED), para el tratamiento de datos de absorbancia total de la mezcla con un número determinado de *Factores* y de *Niveles* para cada sistema cinético, que configuran un *Dominio Experimental* bidimensional o tridimensional dependiendo del modelo. Hemos considerado las variaciones sistemáticas y continuas de los valores de ε_j^λ entre los valores de las configuraciones extremas con relaciones mutuas entre todas las especies. Esto significa abarcar el intervalo completo, lo cual nos va a garantizar que cualquiera de los casos reales que podamos encontrarnos en el laboratorio, está incluido en el *Dominio Experimental* considerado en el *Diseño de Experimentos* (ED).

Cuando se realiza el tratamiento de datos procedentes de cinéticas no isotérmicas, que nos van a permitir la determinación de Parámetros Termodinámicos de Activación (PTA), es necesario conocer los valores de la fracción remanente obtenidos a partir de los valores concentración/tiempo junto con la concentración inicial de la especie correspondiente. El desarrollo e implementación de un adecuado *Diseño Experimental* es crucial para garantizar el éxito en el proceso de diseño de la Arquitectura óptima de la ANN y su *Entrenamiento* para que funcione correctamente. Los *Factores* están formados por el conjunto de parámetros A_{12}/E_{12} y A_{23}/E_{23} , cuando se aplica la Ecuación de Arrhenius, $\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$ y $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ en el caso de que se utilice la Ecuación de Eyring, cuyas *respuestas* son los datos cinéticos no isotérmicos de la base de curvas “*Inputs*” (α_1/t , α_2/t , α_3/t).

Es necesario tener en cuenta 2 variables: a) los valores extremos de ambos *Factores* en cada uno de los cuatro conjuntos que configuran el *Dominio Experimental* y b) los valores relativos de cada par de valores de PTA. Ambas variables deben asegurar que las combinaciones binarias de ambos *Factores* generarán un conjunto de curvas cinéticas que tendrán la suficiente información para garantizar un proceso de *Entrenamiento* óptimo de la red neuronal. Además, el número de niveles de los *Factores* del ED debe ser el adecuado si se quiere evitar tanto un trabajo computacional inútil como grandes diferencias en el espaciado de los valores de las respuestas. Por consiguiente, para optimizar el proceso de formación de las curvas cinéticas de la matriz de “*Inputs*” se debe tener información cinética eficiente y deben

ser distribuidas correctamente de acuerdo con la elección de un *Diseño de Experimentos* adecuado y un *Dominio Experimental* apropiado.

De acuerdo con los resultados del estudio de las funciones $T = f(t)$, se han generado los datos cinéticos no isotérmicos para ambas ecuaciones de manera independiente (Arrhenius y Eyring). De este modo se obtienen los datos de las curvas de la matriz de “Inputs” (organizadas según una distribución ED adecuada) para llevar a cabo el proceso de desarrollo de la Red Neuronal que nos permita la correcta *predicción* de los Parámetros Termodinámicos de Activación (PTA). Teniendo en cuenta las características de computación de las ANN de los sistemas cinéticos no isotérmicos a estudiar, hemos elegido como óptimo el ED correspondiente al tipo *Central Star Composite Experimental Design* (CSCED).

V.2.3.- Tipos de ordenación de las matrices de datos cinéticos isotérmicos y no isotérmicos

En las matrices *Inputs* y en las ordenaciones matriciales de representación gráfica, hemos considerado 2 criterios de ordenación, según se consignan en una misma fila o columna de la matriz, los valores de las variables de una manera secuencial o no secuencial. El orden secuencial, consiste en escribir en una misma fila o columna de la matriz las n (siendo n el número de especies para cada uno de los sistemas) series completas de absorbancias, concentraciones o fracciones remanentes de cada una de las especies que pertenecen a una misma curva, como n bloques consecutivos uno a continuación del otro, sin intercalar valores de cada bloque. Cualquier otra ordenación distinta a ésta, es considerada del tipo no secuencial.

Las representaciones gráficas 3D se realizan de diferente manera dependiendo de la aplicación de software comercial y/o “libre” que se utilice. Nosotros hemos utilizado diferentes aplicaciones, eligiendo en cada caso la que produce una figura más ilustrativa y por tanto más conveniente. Podría pensarse en agrupar los datos en matrices multidimensionales 3D pero puede hacerse de una manera más sencilla: los datos a representar pueden estar contenidos en una matriz 2D con 3 columnas en la que se asigna previamente cada columna a cada eje (X, Y, Z) eligiendo una representación 3D del tipo “*Scatter*” como ocurre en ORIGIN (Figura V.2.3.1). MATLAB es la que más alternativas ofrece, pues los datos pueden agruparse en n vectores columna distintos o matrices monodimensionales de “ n ” elementos, utilizando la función MATLAB “*scatter3*”, o bien en otra forma más general agrupando los datos en una matriz 2D, utilizando n bloques homogéneos en representaciones 3D tipo “*surf*” (opción que nosotros hemos elegido en la Figura V.2.2.2) u otros tipos “*mesh*”, “*contour maps*”, “*ribbon*” “*water fall*”,... etc. Incluso, una gráfica sencilla 2D puede resultar suficiente utilizando incluso una sencilla Hoja de Cálculo EXCEL, como por ejemplo ocurre en la Figura V.2.2.1.

A continuación vamos a describir detalladamente diversas posibilidades de ordenación matricial de curvas y datos cinéticos que hemos utilizado y la evaluación de los resultados obtenidos tras considerar las ventajas y los inconvenientes que presentan. Todos estos casos los hemos recogido, a modo de resumen, en la Tabla V.2.1 y en el APENDICE V.1, donde se expresa la composición detallada de todas las matrices de entrada *Inputs* que han sido sometidas tanto a computación ANN como a representación gráfica. En la mayoría de los casos, hemos considerado como base óptima 45 curvas cinéticas (n_c) que contienen valores de absorbancia total del sistema (A_{T,t_i,n_c}^λ), valores de absorbancia individual de la especie “j”, adquiridos a la longitud de onda λ (A_{j,t_i,n_c}^λ), valores de la fracción remanente individual de la especie “j” (α_{j,t_i,n_c}), o bien valores de concentración de la especie “j” ($[B_j]$) obtenidos a distintos valores de tiempo (t_i). Las dimensiones de las matrices de *Targets* son diferentes dependiendo del sistema tratado y de las variables a optimizar.

Casos 1a, 1b y 1c.-

Denominación: C1a: Cinéticas Isotérmicas (Sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$)

C1b: Cinéticas Isotérmicas (Sistema $B_1 \xrightleftharpoons[k_{21}]{k_{12}} B_2 \xrightarrow{k_{23}} B_3$)

C1c: Cinéticas No-Isotérmicas: Arrhenius (Sistema $B_1 \xrightarrow{k_{12}} B_2$)

Propiedad medida C1a y C1b: Absorbancia total del sistema (A_{T,t_i,n_c}^λ)

Propiedad medida C1c: Fracción remanente de la especie “2” (α_{2,t_i,n_c})

Fila genérica curva “n_c” matriz 2D de “Input” ($A_{T,t_1,n_c}^\lambda \ A_{T,t_2,n_c}^\lambda \ A_{T,t_3,n_c}^\lambda \ \dots \ A_{T,t_{50},n_c}^\lambda$)

Fila genérica curva “n_c” matriz 2D de “Input” ($\alpha_{2,t_1,n_c} \ \alpha_{2,t_2,n_c} \ \alpha_{2,t_3,n_c} \ \dots \ \alpha_{2,t_{50},n_c}$)

Matriz 2D de “Input”: C1a: ($A_{T,t_i,n_c}^\lambda / n_c$) (50x45)

C1b: ($A_{T,t_i,n_c}^\lambda / n_c$) (50x59)

C1c: ($\alpha_{2,t_i,n_c} / n_c$) (50x45)

Computación ANN: C1a: SI (matriz de *Targets*: (2x45))

C1b: SI (matriz de *Targets*: (3x59))

C1c: SI (matriz de *Targets*: (2x45))

Representaciones gráficas 3D (X/Y/Z): C1a: ($t_i / A_{T,t_i,n_c}^\lambda$) (50/50/45)

C1b: ($t_i / A_{T,t_i,n_c}^\lambda$) (50/50/59)

C1c: ($n_c / \alpha_{2,t_i,n_c}$) (45/50/50)

Representaciones 3D: MATLAB

Tipo, forma y Figuras: NO Secuencial, “*toboganes múltiples*”, (Figura V.2.1.a, Figura V.2.1.b; otros, Figura V.2.1.c.)

→ Ver APENDICE V.1 y Tabla V.2.1

Comentarios:

La gran mayoría de las figuras, correspondientes a los sistemas que hemos estudiado son muy similares por lo que mostramos únicamente 3 ejemplos, (2 para el caso de cinéticas isotérmicas y 1 para cinéticas no isotérmicas). Los casos que estamos considerando (1a, 1b y 1c) son quizás de más importantes y los que con mayor frecuencia hemos utilizado tanto en

representación gráfica como en la propia computación ANN. Poseen innegables ventajas, pues los valores de absorbancia total y de fracción remanente son datos experimentales directamente adquiridos de su monitorización “*on line*”. En el caso de la absorbancia total contribuyen simultáneamente todas las especies y sus valores poseen un amplio intervalo de variación (entre 0.10 y 1.10 aprox.).

En cinéticas isotérmicas, las figuras 3D responden a muy diversos perfiles (Figuras V.2.1.a y V.2.1.b) pues dependen de la amplia casuística de los valores relativos de los 3 coeficientes de absorción molar de las 3 especies. En general, presentan suaves perfiles de variación sin discontinuidades ni abruptas diferencias. Sin embargo, tiene algunas desventajas derivadas de los valores relativos de los coeficientes de absorción molar en relación a la selectividad de la longitud de onda de trabajo (ya comentado en el apartado V.2.2) y que da lugar a problemas que pueden hacer inviable su tratamiento. Cuando los valores de los ε_j^{λ} de las 3 especies sean muy parecidos con solapamiento de bandas de las 3 especies, el intervalo de variación de los valores de la absorbancia total de la mezcla será mínimo e insuficiente para que las técnicas ANN puedan producir buenos resultados. Algo similar ocurriría en el caso en que 2 valores de ε_j^{λ} correspondientes a 2 especies, tuvieran valores iguales o similares y el correspondiente a la tercera especie, fuera nulo o despreciable frente a los otros. No hay que olvidar que estos casos son extremos y son los que corresponden a los puntos ubicados en los vértices de las superficies o los volúmenes de los Diseños de Experimentos (ED) pero que es necesario incluir en la etapa de *Entrenamiento* de la red neuronal como casos límite. Estos son 2 supuestos en los que no es posible el tratamiento computacional ANN por lo que habría que recurrir a otros tipos de datos cinéticos (absorbancias y/o concentraciones individuales) y que vamos a considerar posteriormente en los otros casos.

En cinéticas no isotérmicas, las circunstancias son muy similares a los casos anteriores. Se puede observar que las figuras poseen un gran parecido entre ellas, debido a que el tiempo, la fracción remanente y/o la absorbancia total poseen los mismos intervalos de variación en la mayoría de los casos. Una gran ventaja con respecto a las cinéticas isotérmicas es trabajar con las ecuaciones de Arrhenius y Eyring, ya que no es necesario considerar los coeficientes de absorción molar de ninguna de las especies, que hace que el tratamiento ANN sea más sencillo. Sin embargo, la gran diferencia en los órdenes de magnitud entre los parámetros termodinámicos de activación a determinar, hace que la computación con ANN sea mucho más tediosa y genere problemas de tratamiento que comentaremos más adelante en detalle.

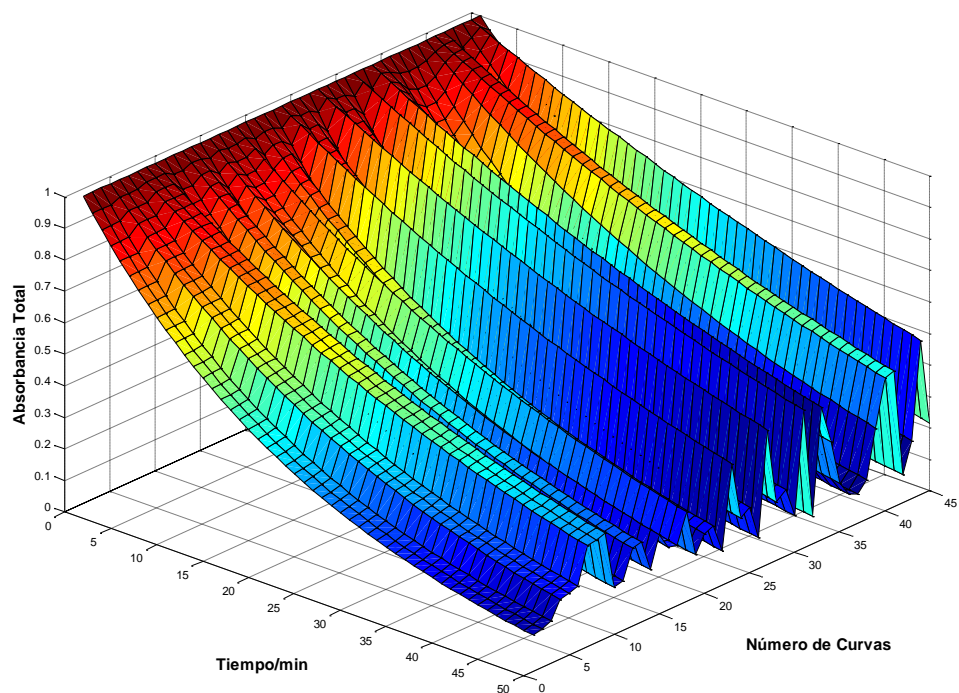


Figura V.2.1.a: Representación gráfica en 3D del sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ para cinéticas isotérmicas.

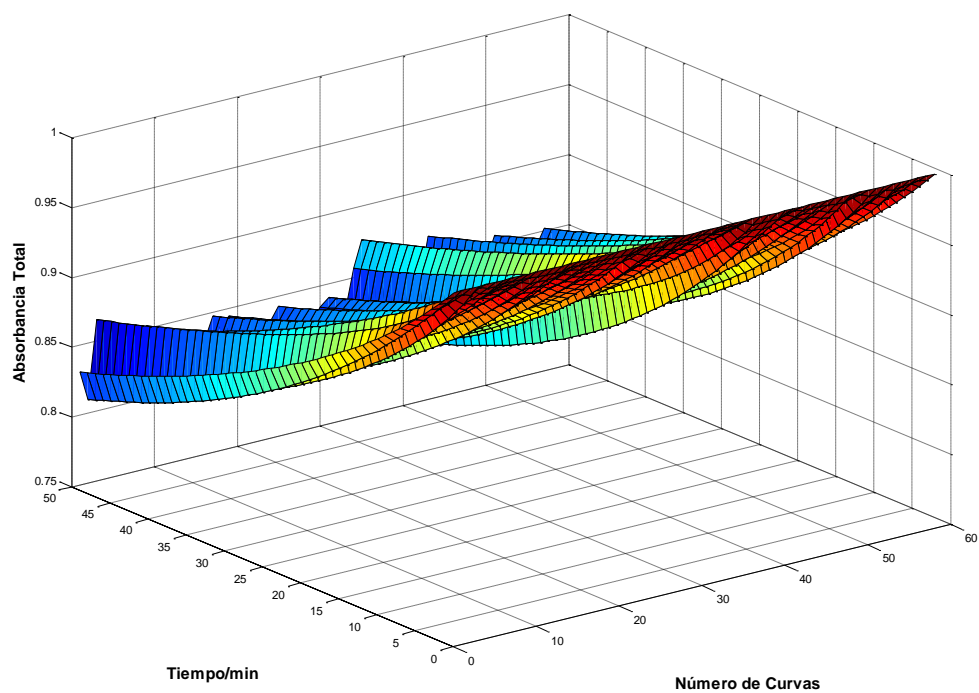


Figura V.2.1.b: Representación gráfica en 3D del sistema $B_1 \xrightleftharpoons[k_{21}]{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ para cinéticas isotérmicas.

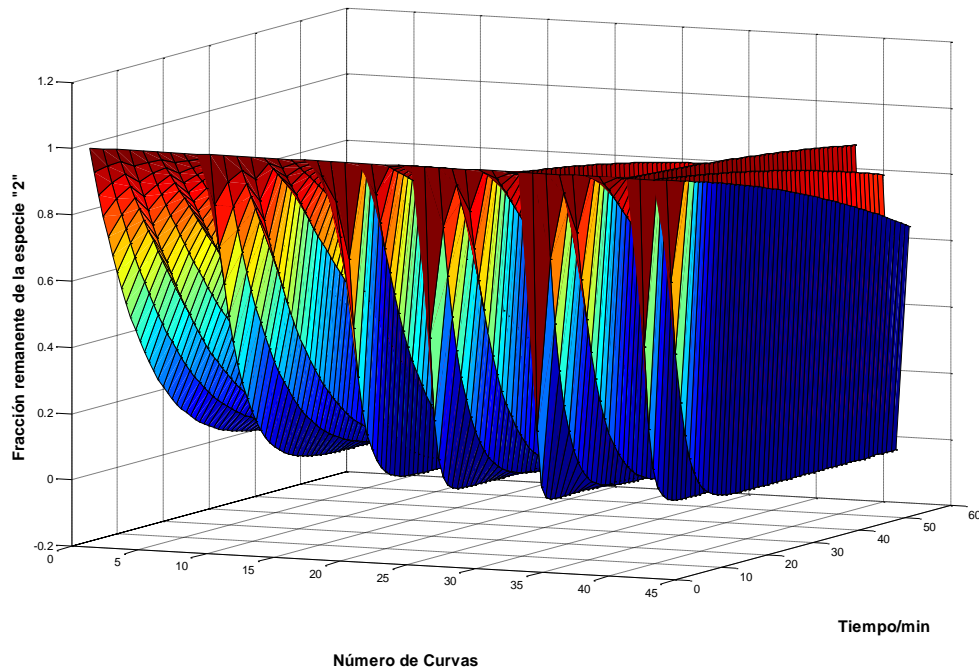


Figura V.2.1.c: Representación gráfica en 3D del sistema $B_1 \xrightarrow{k_{12}} B_2$ para cinéticas no isotérmicas: Arrhenius.

Casos 2a, 2b y 2c.-

Denominaciones: C2a y C2b: Cinéticas Isotérmicas (**Sistema** $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$)

C2c: Cinéticas No-Isotérmicas (**Sistema** $B_1 \xrightarrow{k_{12}} B_2$)

Propiedad medida C2a y C2b: concentración de cada especie “j” ($[B_j]_{t_i, n_c}$)

Propiedad medida C2c: SQD ($SQD_{j, A_{12}, E_{12}}$)

Fila genérica matriz “Input” curva “n_c” (concentraciones):

$[B_1]_{t_1, n_c} \cdots [B_1]_{t_{50}, n_c} [B_2]_{t_1, n_c} \cdots [B_2]_{t_{50}, n_c} [B_3]_{t_1, n_c} \cdots [B_3]_{t_{50}, n_c}$

Fila genérica matriz “Input” (SQD):

$SQD_{1, A_{12}, E_{12}} SQD_{2, A_{12}, E_{12}} SQD_{3, A_{12}, E_{12}} \cdots SQD_{41, A_{12}, E_{12}}$

Matriz 2D de “Input”: C2a y C2b: ($[B]_{(1,2,3), t_i / n_c}$) (45x150)

Computación ANN: C2a y C2b: SI (matriz *Targets*: 45x2)

Representación gráfica 2D matriz “Input” [X/Y]: C2a: $[B_{(1,2,3)}]_{t_1, n_c} / 3(t_i)$

Representaciones gráficas 2D: C2a: EXCEL, (Figura V.2.2.1) (45x150)

Tipo, Forma y Figura: C2a: “Secuencial sin Solapamiento”; “sillón ergonómico 2D”, (Figura V.2.2.1)

Representación gráfica 3D matriz “Input” [X/Y/Z]: C2b: $[B_{(1,2,3)}]_{t_1, n_c} / 3(t_i) / n_c$

Representaciones gráficas 3D: C2b: MATLAB, (Figura V.2.2.2) (45x150x150)

Tipo, Forma y Figura: C2b: “Secuencial sin Solapamiento”; “sillón ergonómico 3D”, (Figura V.2.2.2)

Representación gráfica 3D matriz “Input” [X/Y/Z]: C2c: $[SQD_{j, A_{12}, E_{12}} / E_{12} / A_{12}]$

Representaciones gráficas 3D: C2c: MATLAB, (Figura V.2.2.3) (41x41x41)

Tipo, Forma y Figura: C2c: “Secuencial sin Solapamiento” (Figura V.2.2.3)

➔ Ver APENDICE V.1 y Tabla V.2.1

Comentarios:

Debido a la gran similitud entre las figuras para los diversos sistemas, hemos tomado como referencia para el tratamiento de Cinéticas Isotérmicas el modelo de reacciones consecutivas con 3 especies y 2 constantes de velocidad. Los elementos de las matrices están distribuidos de acuerdo a una ordenación *Secuencial* cuyas características ya las hemos comentado con anterioridad. Las matrices permiten realizar tanto la representación gráfica 3D como la computación ANN ya que, en este caso particular, las dimensiones de la matriz de *Inputs* (C2a y C2b) son 45×150 y las de la matriz de *Targets* son 45×2 .

Aunque los valores individuales de absorbancias y concentraciones no se pueden medir directamente, resulta muy útil su computación ANN fundamentalmente cuando no pueden aplicarse otros tipos de ordenación matricial de curvas y datos cinéticos (Caso C1a) pero se conocen los valores de $\varepsilon_j^{\lambda_j}$. En un análisis superficial podría parecer que en computación ANN el uso de absorbancias o concentraciones individuales de cada especie, no debería presentar diferencias significativas ya que se trata de valores proporcionales entre si y supondría un simple cambio de escala. Sin embargo los perfiles resultantes de las curvas de concentraciones y de absorbancias, en general son completamente diferentes puesto que están constituidas por sumas de términos cruzados. De acuerdo con esta observación, hay que considerar siempre que sea posible, la doble computación ANN con datos de ambos tipos. En la mayoría de los ocasiones suelen coincidir los resultados obtenidos al computar datos procedentes de ambos casos y el estudio sería redundante. Si no existe concordancia, nos quedaremos con aquel que produzca los mejores resultados derivados de los tratamientos Estadísticos posteriores.

La Figura V.2.2.1 trazada mediante la aplicación de la Hoja de Cálculo EXCEL, corresponde a la representación 2D de los elementos de la matriz de *Input*. Es decir, son los valores de concentración de cada especie en cada curva, frente a sus correspondientes valores de tiempo. Los puntos están agrupados en bloques secuenciales consecutivos de acuerdo al orden establecido en el mecanismo ($B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$). La Figura V.2.2.2 corresponde a la representación gráfica 3D trazada con MATLAB con valores exclusivamente de concentraciones. Se observan perfiles típicos y fácilmente reconocibles ya que presentan 3 partes bien diferenciadas correspondientes a los perfiles 3D de cada una de las 3 especies secuencialmente ordenadas, acordes al mecanismo de reacciones consecutivas ($B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$). Hemos realizado representaciones gráficas 2D y 3D en concentraciones (casos C2a y C2b) porque en el caso de absorbancias individuales de cada especie, los perfiles de las figuras resultantes dependen de los valores relativos de los 3 valores de los coeficientes de absorción molar. Para un determinado conjunto de valores de $\varepsilon_j^{\lambda_j}$, pueden reconocerse las 3 partes

homogéneas con diferentes perfiles que obedecen a la representación del denominador tipo *Secuencial*.

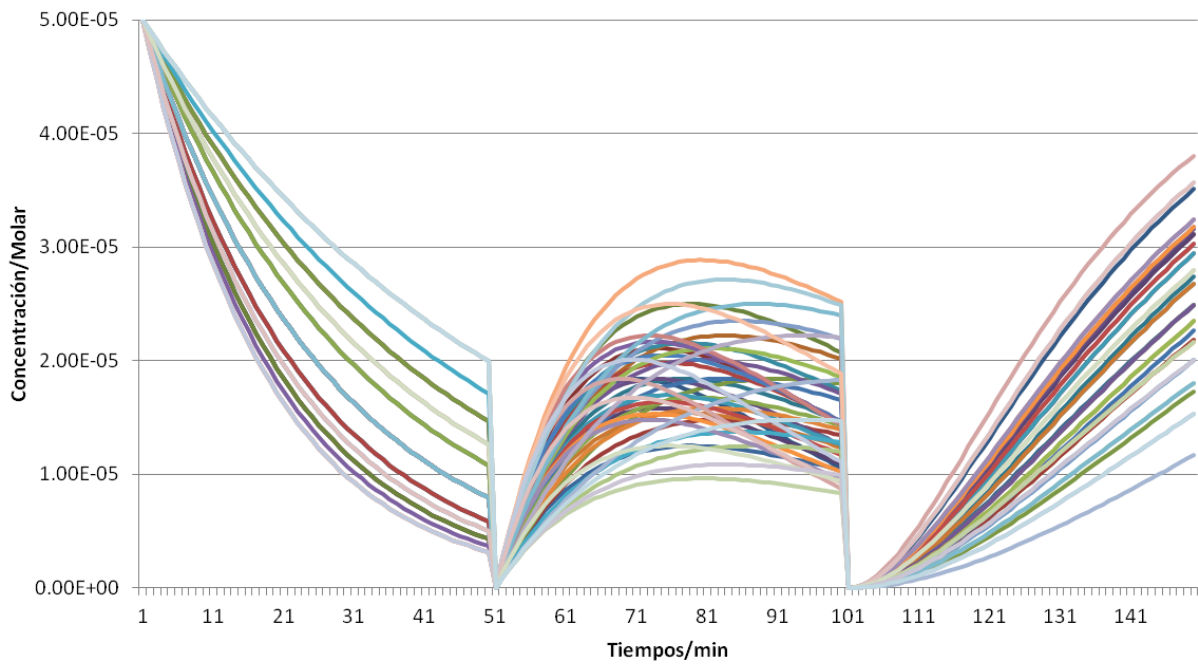


Figura V.2.2.1: Representación gráfica en 2D del sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ para los elementos de la matriz de *Input*.

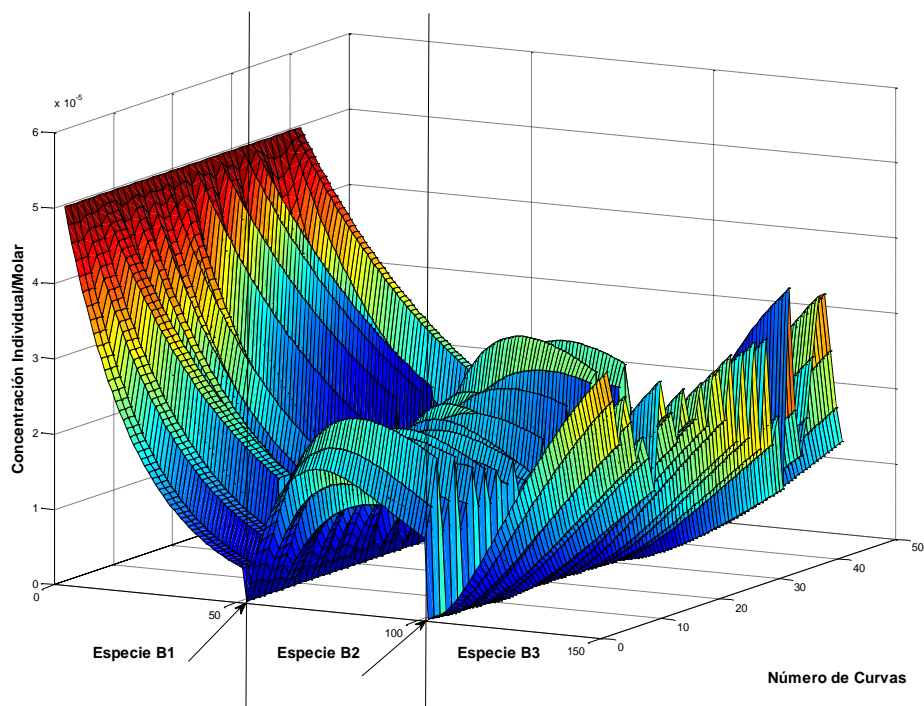


Figura V.2.2.2: Representación gráfica en 3D del sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ con valores de concentraciones individuales.

En el caso de la determinación de Parámetros Termodinámicos de Activación (PTA) a partir de datos procedentes de Cinéticas No Isotérmicas, hemos utilizado un Algoritmo Híbrido (AH) que combina 2 metodologías diferentes: ANN y AGDC. El algoritmo AGDC minimiza la función suma de desviaciones cuadráticas (SQD) por lo que es muy conveniente analizar gráficamente los perfiles de los mínimos de la función SQD dependiente de los PTA. Este análisis permite conocer *a priori* si existen en el entorno del mínimo puntos singulares (máximos y/o mínimos locales, “*saddle points*”, etc) donde se anula el vector gradiente que conduciría al fracaso del proceso de optimización. Puesto que la función SQD es una función numérica, hemos realizado una representación gráfica 3D generando un gran colectivo de curvas no isotérmicas a partir de pares de valores de los Parámetros Termodinámicos de Activación distribuidos de acuerdo al ED elegido en las proximidades del mínimo, que corresponde al “*punto central*”. La representación gráfica 3D (Figura V.2.2.3) muestra la evolución hacia el mínimo de la función SQD.

Tras realizar giros adecuados de los ejes para obtener la mejor perspectiva de visualización, no se observa ningún punto singular en las inmediaciones del mínimo. .

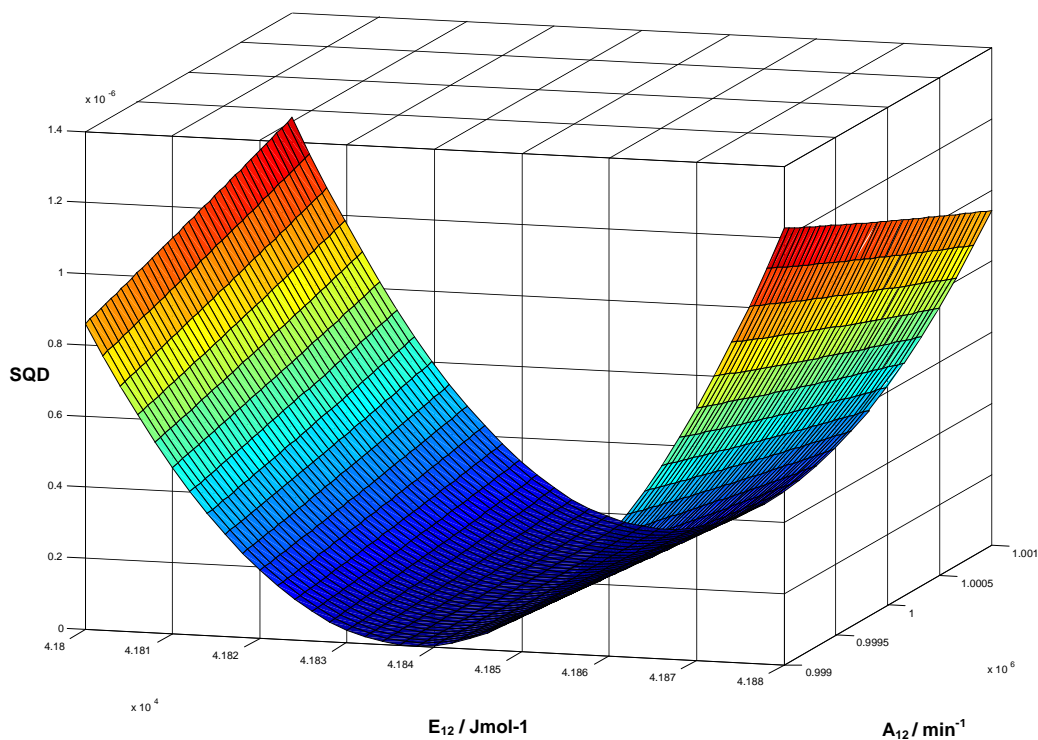


Figura V.2.2.3: Representación gráfica en 3D del sistema $B_1 \xrightarrow{k_{12}} B_2$ de la función numérica SQD vs A y E mostrando el mínimo de la función de ambos parámetros.

Casos 3a y 3b.-

Denominaciones: C3a y C3b: Cinéticas Isotérmicas (*Válido para sistemas con 3 especies*)

Propiedades medidas:

C3a: concentración de cada especie “j” ($[B_j]_{t_i, n_c}$)

C3b: concentración de cada especie “j” ($[B_j]_{t_i, n_c}$)

Fila genérica matriz “Input” curva “n_c” (concentraciones): ($[B_1]_{t_1, n_c}$ $[B_2]_{t_1, n_c}$ $[B_3]_{t_1, n_c}$)

Representación gráfica 3D [X/Y/Z]: ($[B_1]_{t_i}$ / $[B_2]_{t_i}$ / $[B_3]_{t_i}$) (2250x2250x2250)

Dimensiones matriz 2D de “Input”: (2250x3)

Computación ANN: NO ES POSIBLE en el caso $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ (matriz *Targets* (45x2))

Representación 3D: MATLAB(*) y ORIGIN (Figuras V.2.3.1 y V.2.3.2)

Tipo, Forma y Figuras: No Secuencial; “Banana”, Figuras V.2.3.1 y V.2.3.2

→ Ver APENDICE V.1 y Tabla V.2.1

Comentarios

a) **Representación gráfica:** Las figuras de representación gráfica 3D son muy ilustrativas y poseen muchas ventajas (Figuras V.2.3.1 y V.2.3.2) ya que proceden de la representación de muchos puntos y configuran una forma característica y fácilmente identificable tipo “banana”. Sin embargo, esta representación resulta extraña que para una representación típica de datos cinéticos. Lo que ocurre es que la nueva variable independiente pasa a ser el número de especies ($n_j = 3$) lo cual resulta especialmente apropiado en una representación 3D para muchos de los sistemas cinéticos en estudio, ya que existen justamente 3 especies con 2 ó 3 constantes cinéticas. Esto supone una gran ventaja, pues permite que en una misma figura puedan superponerse todos los posibles sistemas que posean 3 especies, independientemente tanto del número de constantes cinéticas involucradas ($k_{jj'}$) como de su arquitectura lineal o cíclica., mostrando en el esquema V.2.1 los sistemas con 3 especies trabajados en la presente Memoria. Así, es posible recopilar en una única figura los datos de todos los modelos y la tarea de comparación visual es mucho más sencilla. Además, pueden detectarse tendencias de variación, desviaciones y comportamientos extraños de puntos “outlayers”, errores sistemáticos de medida,...etc., que NO podrían detectarse con los otros tipos de representaciones gráficas.

Número	MECANISMO	Número	MECANISMO
1	$B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$	3	$B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$
2	$B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ $B_1 \xleftarrow{k_{21}}$	4	$B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3 \xrightarrow{k_{31}} B_1$

Esquema V.2.1: Tabla de una serie de mecanismos de reacción seleccionados en el estudio del presente trabajo.

La Figura V.2.3.1 muestra los perfiles de las superficies de representación mientras que en la Figura V.2.3.2 se observa una peculiaridad importante de esta representación gráfica ya que todos los puntos configuran y pertenecen a un plano diagonal cuando se trata de datos sintéticos sin error. Esto supone otra ventaja adicional, pues las desviaciones de los puntos correspondientes a los datos experimentales respecto a ese plano diagonal nos van a indicar de una forma “semicuantitativa” los errores intrínsecos de medida de dichos datos.

b) Computación ANN: Como no coinciden las dimensiones de las matrices de *Input* (2250×3) y *Targets* (45×2) para el sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ en ambas ordenaciones C3a y C3b, NO es posible realizar el tratamiento ANN. Solo en el caso particular de un sistema cinético formado por 3 especies y 3 constantes cinéticas, la matriz de *Targets* tendría dimensiones (45×3) con lo que habría compatibilidad respecto a las dimensiones de ambas matrices. Entonces sería posible aplicar el tratamiento ANN, tal como hemos hecho en otros casos que explicaremos posteriormente. Sin embargo y a pesar de esta coincidencia en las dimensiones de ambas matrices, hemos constatado que los resultados que se obtienen en la fase de *Entrenamiento o Aprendizaje* son muy deficientes. Es decir, resulta viable la aplicación de ANN pero produce resultados inaceptables, que exige un tratamiento posterior de los datos aplicando el algoritmo AGDC. Solo de esta manera se obtienen resultados satisfactorios.

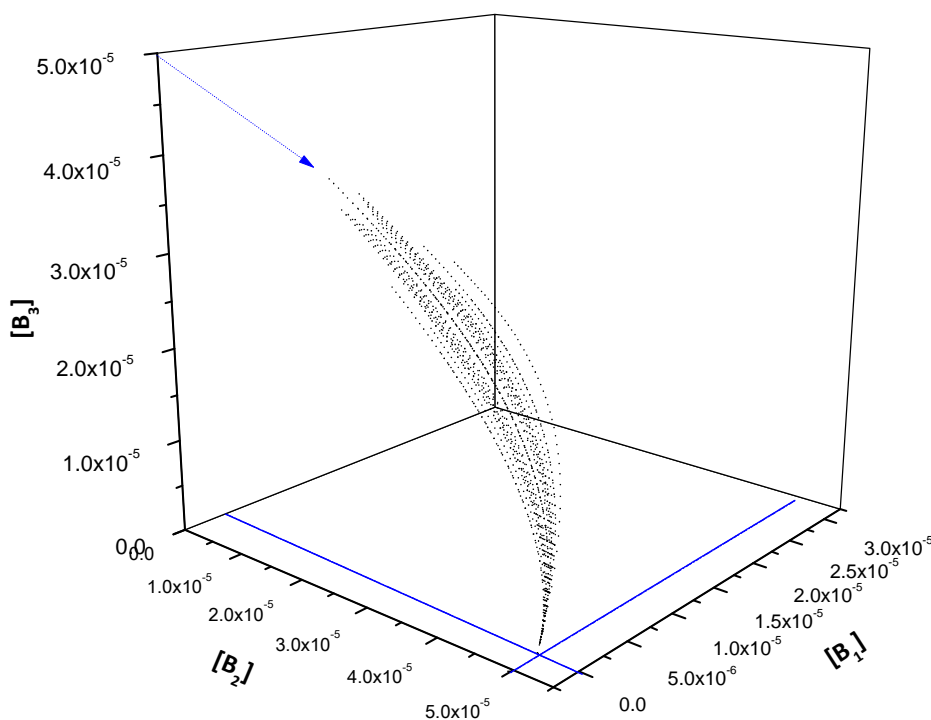


Figura V.2.3.1: Representación gráfica en 3D con forma tipo “banana” para el modelo de reacciones consecutivas.

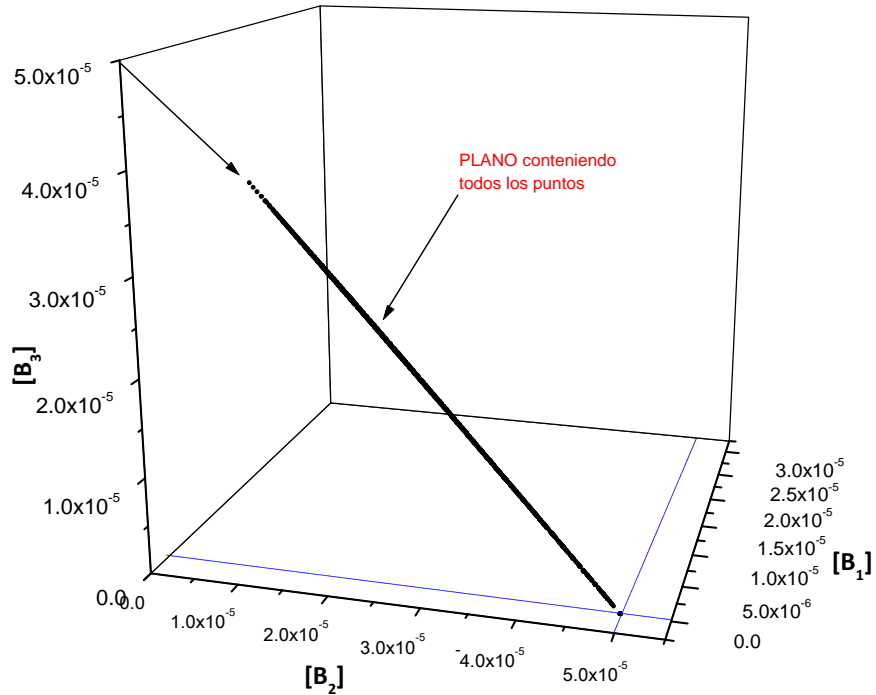


Figura V.2.3.2: Representación gráfica en 3D donde todos los puntos pertenecen a un plano diagonal.

Casos 4a, 4b y 4c.-

Denominaciones: C4a: Cinéticas Isotérmicas (*Sistema* $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$)

C4b: Cinéticas Isotérmicas (*Sistemas* $B_1 \xrightleftharpoons[k_{21}]{k_{12}} B_2 \xrightarrow{k_{23}} B_3, B_1 \xrightarrow{k_{12}} B_2 \xleftarrow[k_{32}]{k_{23}} B_3$ y

$B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3 \xrightarrow{k_{31}} B_1$)

C4c: Cinéticas No-Isotérmicas: Eyring (*Sistema* $B_1 \xrightarrow{k_{12}} B_2$)

Propiedades medidas:

C4a: Concentración de cada especie “j” ($[B_j]_{t_i, n_c}$)

C4b: Absorbancia de cada especie “j” para 3 sistemas distintos ($A_{j, t_i, n_c}^{S_1, S_2, S_3}$)

C4c: Fracción remanente de la especie “2” (α_{2, t_i, n_c})

Fila genérica matriz “Input” tiempo “t_i” y curvas “n_c” (concentraciones, absorbancias y fracción remanente):

C4a: $[B_1]_{t_i, 1} [B_2]_{t_i, 1} [B_3]_{t_i, 1} [B_1]_{t_i, 2} [B_2]_{t_i, 2} [B_3]_{t_i, 2} \dots [B_1]_{t_i, 45} [B_2]_{t_i, 45} [B_3]_{t_i, 45}$

C4b: $A_{1, t_i, 1}^{S_1} A_{2, t_i, 1}^{S_1} A_{3, t_i, 1}^{S_1} A_{1, t_i, 1}^{S_2} A_{2, t_i, 1}^{S_2} A_{3, t_i, 1}^{S_2} A_{1, t_i, 1}^{S_3} A_{2, t_i, 1}^{S_3} A_{3, t_i, 1}^{S_3}$

C4c: $\alpha_{2, t_i, 1} \alpha_{2, t_i, 2} \alpha_{2, t_i, 3} \dots \alpha_{2, t_i, 45}$

Matriz 2D de “Input” C4a: ($[B_j]_{(1,2,3), n_c} / t_i$); (135x50)

C4b: ($A_{(1,2,3), n_c}^{S_1, S_2, S_3} / t_i$); (9x50)

C4c: ($\alpha_{2, t_i, n_c} / t_i$) (45x25)

Computación ANN: NO

Representación gráfica 2D matriz “Input” [X/Y]: C4a: $(t_i / ([B_j]_{(1,2,3),n_c}); (50 \times 135)$

C4b: $(t_i / (A_{(1,2,3),n_c}^{S_1, S_2, S_3}); (50 \times 9)$

C4c: $(t_i / \alpha_{2,t_i,n_c}); (25 \times 45)$

Representaciones gráficas 2D: MATLAB; (50x135) Figura V.2.4.a

MATLAB; (50x9) Figura V.2.4.b

MATLAB; (25x45) Figura V.2.4.c

Tipo, y Figura: “Secuencial con Solapamiento”, Figura V.2.4.a, Figura V.2.4.b y Figura V.2.4.c.

→ Ver APENDICE V.1 y Tabla V.2.1

Comentarios:

La Figura V.2.4.a constituye una variante de los casos 3a y 3b. La diferencia más notable es que la figura resultante de la representación gráfica 2D de las matrices de *Input* es una superposición de los perfiles de las concentraciones de las 3 especies. La ordenación matricial es *Secuencial* pero en la coordenada “Y” las concentraciones de cada una de las 3 especies están agrupadas por bloques de curvas ($n_c = [1, 45]$) en lugar de bloques de tiempos ($t_i = [1, 50]$). La coordenada “X” corresponde al tiempo (t_i) por lo que resulta una gráfica 2D en el que las curvas de evolución temporal de cada especie se solapan mutuamente. Como las dimensiones de las matrices de *Input* y de *Targets* son incompatibles, resulta inviable el tratamiento computacional ANN.

En la Figura V.2.4.b se realiza la representación gráfica 2D de las matrices *Input* (superposición de los perfiles de las absorbancias individuales de las tres especies), en el caso de los 3 sistemas considerados y para una única curva (punto central del ED). La representación de un número más elevado de curvas en la misma figura generaría un grave problema debido a la incapacidad de diferenciar un sistema de otro dada la gran similitud entre los 3 modelos cinéticos. La coordenada “Y” corresponde a las absorbancias de cada una de las 3 especies para cada sistema y la coordenada “X” al tiempo (t_i) por lo que el resultado es una representación en 2D en el que podemos observar la evolución temporal de cada especie y para cada sistema pudiendo comprobar que los perfiles son muy similares en las 3 situaciones.

Por último, la Figura V.2.4.c muestra los perfiles de las curvas cinéticas no isotérmicas (α_2/t). Se puede observar claramente la presencia de varios tramos cóncavos mientras que las curvas isotérmicas son siempre convexas. Este hecho se explica porque en las cinéticas no isotérmicas existen dos efectos simultáneos opuestos que afectan a la velocidad de la reacción: el incremento de la temperatura con el tiempo que aumenta la velocidad de reacción y la disminución de la concentración de reactivo que se va agotando, que disminuye la velocidad. De acuerdo con lo expuesto, se observan los siguientes perfiles: si la influencia de estos efectos opuestos es idéntica, da lugar a una recta. Si predomina el efecto de la concentración sobre el efecto de la temperatura se producirá una curva convexa y si existe una predominancia del efecto de la temperatura sobre el efecto de la concentración se producirá una curva cóncava.

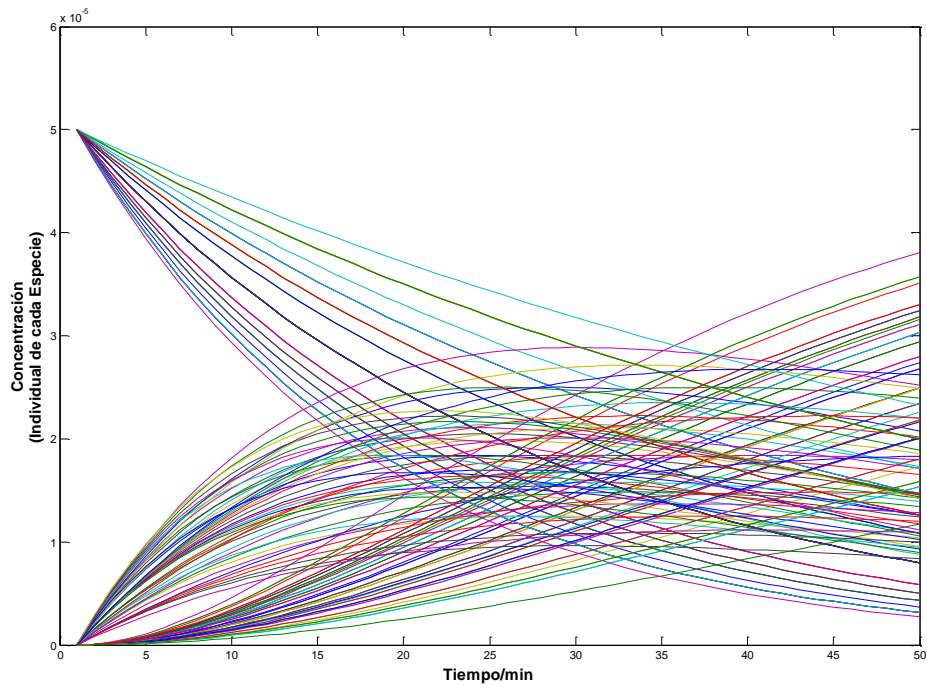


Figura V.2.4.a: Representación gráfica 2D del sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ donde las curvas de cada especie solapan.

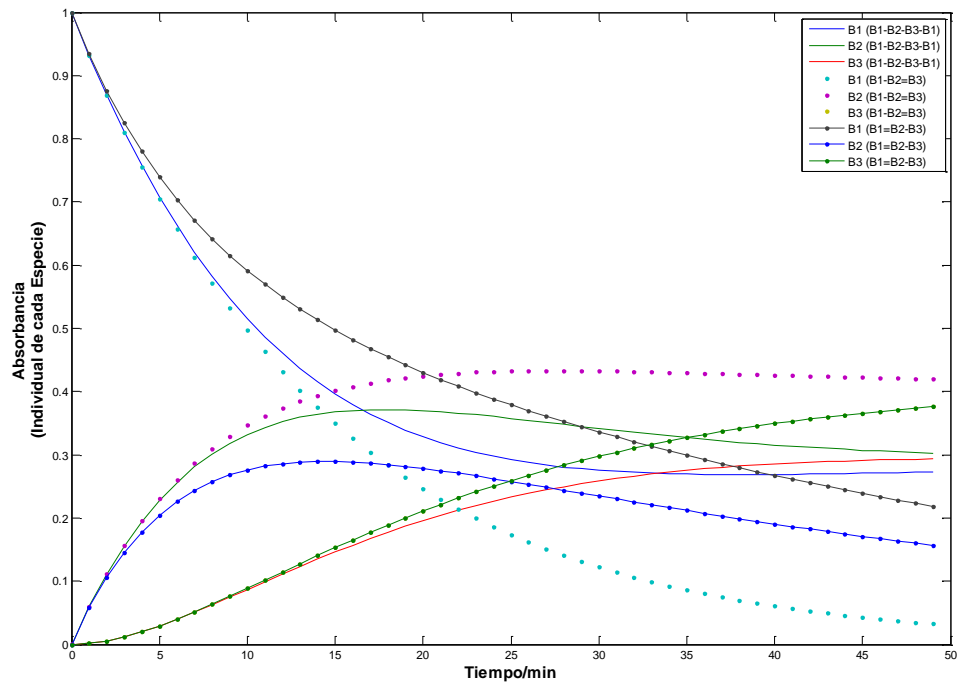


Figura V.2.4.b: Representación gráfica 2D donde se observa la superposición de las absorbancias para los 3 sistemas.

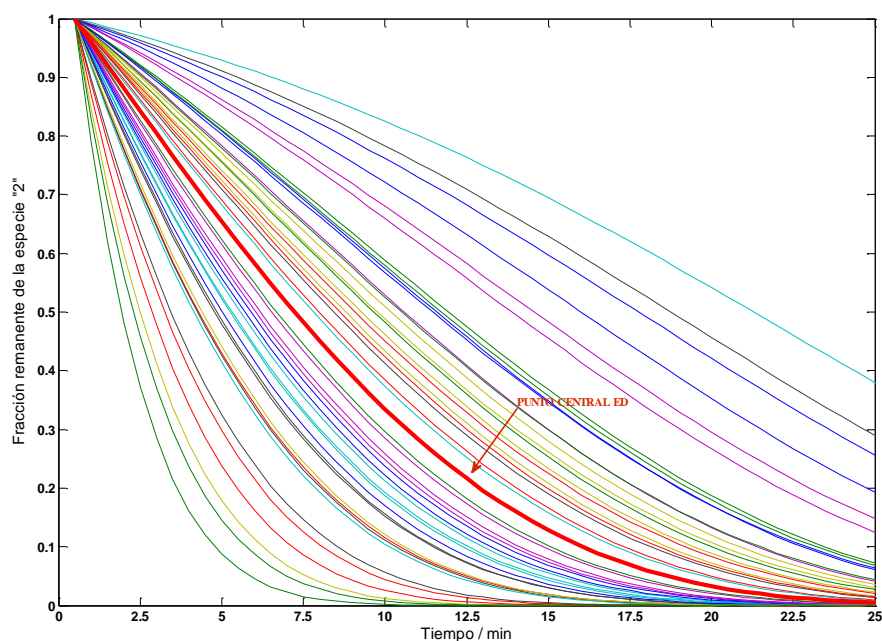


Figura V.2.4.c: Representación gráfica 2D del sistema $B_1 \xrightarrow{k_{12}} B_2$ de los perfiles de las curvas no isotérmicas.

SIMBOLOGÍA y ABREVIATURAS utilizadas en la Tabla V.2.1.

a) SIMBOLOGÍA

Variable	Símbolo	Intervalo Variación	Observaciones
Tiempos	t_i	$0 \leq t_i \leq 50$ [0,50]	Número de tiempos: t_i
Especies	B_j	$1 \leq j \leq 3$	Subíndice (B_1 , B_2 y B_3)
Número de curvas cinéticas	n_c	$1 \leq n_c \leq 59$	Base de curvas: n_c
Concentración de las especies	$[B_j]$	$1 \leq j \leq 3$	
Concentración inicial	$[B_j]_0$	—	$0.5 \cdot 10^{-4} \text{ mol L}^{-1} (\text{dm}^{-3})$
Absorbancia total	A_T	$0 \leq A_T \leq 1.50$	Normalmente < 1.0
Absorbancia individual	A_j	$0 \leq A_j \leq 1.50$	Normalmente < 1.0
Longitud de onda de trabajo	λ	$\lambda = 350 \text{ nm}$ (p.ej)	superíndice
Fracción remanente	α_j	$0 \leq \alpha_j \leq 1$	Normalmente < 1.0

b) ABREVIATURAS

MAT = MATLAB
 TRAJ = TRAJAN
 ORI = Hoja de Cálculo ORIGIN
 EXC = Hoja de Cálculo EXCEL
 QTIP = QTIPLLOT (software libre)
 Sec = Ordenación Secuencial
 NOsec = Ordenación NO Secuencial
 Tobogán = forma de "toboganes múltiples" (término coloquial)
 Sillón = forma de "sillón ergonómico" (término coloquial)
 Banana = forma de "banana" (término coloquial)

Tabla V.2.1. Resumen de los tipos de ordenación de las matrices y sus representaciones (*)
 (*) Ver los cuadros de Simbología y Abreviaturas de Tabla V-2-1

		REPRESENTACIONES GRÁFICAS 2D y 3D					COMPUTACIÓN REDES NEURONALES ARTIFICIALES (ANN)				
VARIABLES	Propiedad	Ejes X/Y/Z	Dimensión matriz 3D/2D	Tipo	Software Aplicación Informática	Figura	Forma	Elementos (Propiedad)	Dimensión matriz Input 2D	Software Aplicac Informát	
C1a y C1b	A_{T,t_i,n_c}^2	$A_{T,t_i,n_c}^2/t_i/n_c$	50x50x45 50x50x59	NOSec	MAT	Fig. V.2.1a y b	Tobogan (otros)	$A_{T,t_i,n_c}^2/n_c$	50x45 50x59	MAT TRAJ	
C1c	α_{2,t_i,n_c}	$\alpha_{2,t_i,n_c}/t_i/n_c$	50x50x45	NOSec	MAT	Fig. V.2.1c		$\alpha_{2,t_i,n_c}/n_c$	50x45	MAT	
C2a	$[B_j]_{t_i,n_c}$	$[B_{(1,2,3)}]_{t_1,n_c}/3(t_i)$	45x150 (2D)	Sec	EXC	Fig. V.2.2.1		$[B_{(1,2,3)}]_{t_i,n_c}/n_c$	45x150	MAT TRAJ	
C2b	$[B_j]_{t_i,n_c}$	$[B_{(1,2,3)}]_{t_1,n_c}/3(t_i)/n_c$	45x150x150	Sec	MAT	Fig. V.2.2.2	Sillón	$[B_{(1,2,3)}]_{t_i,n_c}/n_c$	45x150	MAT TRAJ	
C2c	$SQD_{j,A_{12},E_{12}}$	$[SQD_{j,A_{12},E_{12}}/E_{12}/A_{12}]$	41x41x41	Sec	MAT	Fig. V.2.2.3		$SQD_{j,A_{12},E_{12}}/E_{12}$	41x41	MAT TRAJ	
C3a y C3b	$[B_j]_{t_i,n_c}$	$[B_1]_{t_i,n_c}/[B_2]_{t_i,n_c}/[B_3]_{t_i,n_c}$	2250x 2250x2250	NOSec	ORI	Fig. V.2.3.1 y V.2.3.2	Banana	$[B_j]_{t_i,n_c}/n_j$	3x2250	NO	
C4a	$[B_j]_{t_i,n_c}$	$[B_{(1,2,3)}]_{t_1,n_c}/t_i$	135x50 (2D)	Solapamiento	MAT	Fig. V.2.4.a	2D	$[B_{(1,2,3)}]_{t_1,n_c}/t_i$	135x50	NO	
C4b	$A_{j,t_i,n_c}^{S_1,S_2,S_3}$	$A_{(1,2,3),n_c}^{S_1,S_2,S_3}/t_i$	9x50 (2D)	Solapamiento	MAT	Fig. V.2.4.b	2D	$A_{(1,2,3),n_c}^{S_1,S_2,S_3}/t_i$	9X50	NO	
C4c	α_{2,t_i,n_c}	$\alpha_{2,t_i,n_c}/t_i$	45x25 (2D)	Sec	MAT	Fig. V.2.4.c	2D	$\alpha_{2,t_i,n_c}/t_i$	45X25	NO	

APENDICE V-1.

→ **Expresiones detalladas de las matrices *Input* en los distintos casos:**

Casos C1a, C1b y C1c

C1a

$$\begin{pmatrix} A_{T,t_1,1}^\lambda & A_{T,t_2,1}^\lambda & \dots & A_{T,t_{50},1}^\lambda \\ A_{T,t_1,2}^\lambda & A_{T,t_2,2}^\lambda & \dots & A_{T,t_{50},2}^\lambda \\ \vdots & \vdots & \ddots & \vdots \\ A_{T,t_1,45}^\lambda & A_{T,t_2,45}^\lambda & \dots & A_{T,t_{50},45}^\lambda \end{pmatrix}$$

C1b

$$\begin{pmatrix} A_{T,t_1,1}^\lambda & A_{T,t_2,1}^\lambda & \dots & A_{T,t_{50},1}^\lambda \\ A_{T,t_1,2}^\lambda & A_{T,t_2,2}^\lambda & \dots & A_{T,t_{50},2}^\lambda \\ \vdots & \vdots & \ddots & \vdots \\ A_{T,t_1,45}^\lambda & A_{T,t_2,45}^\lambda & \dots & A_{T,t_{50},59}^\lambda \end{pmatrix}$$

C1c

$$\begin{pmatrix} \alpha_{2,t_1,1} & \alpha_{2,t_2,1} & \dots & \alpha_{2,t_{50},1} \\ \alpha_{2,t_1,2} & \alpha_{2,t_2,2} & \dots & \alpha_{2,t_{50},2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{2,t_1,45} & \alpha_{2,t_2,45} & \dots & \alpha_{2,t_{50},45} \end{pmatrix}$$

Casos C2a, C2b y C2c

C2a y C2b

$$\begin{pmatrix} [B_1]_{t_1,1} & \dots & [B_1]_{t_{50},1} & [B_2]_{t_1,1} & \dots & [B_2]_{t_{50},1} & [B_3]_{t_1,1} & \dots & [B_3]_{t_{50},1} \\ [B_1]_{t_1,2} & \dots & [B_1]_{t_{50},2} & [B_2]_{t_1,2} & \dots & [B_2]_{t_{50},2} & [B_3]_{t_1,2} & \dots & [B_3]_{t_{50},2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ [B_1]_{t_1,45} & \dots & [B_1]_{t_{50},45} & [B_2]_{t_1,45} & \dots & [B_2]_{t_{50},45} & [B_3]_{t_1,45} & \dots & [B_3]_{t_{50},45} \end{pmatrix}$$

C2c

$$\begin{pmatrix} SQD_{1,E_{12},A_{12}} & SQD_{2,E_{12},A_{12}} & \cdots & SQD_{41,E_{12},A_{12}} \\ SQD_{1,E_{12},A_{12}} & SQD_{2,E_{12},A_{12}} & \cdots & SQD_{41,E_{12},A_{12}} \\ \vdots & \vdots & \ddots & \vdots \\ SQD_{1,E_{12},A_{12}} & SQD_{2,E_{12},A_{12}} & \cdots & SQD_{41,E_{12},A_{12}} \end{pmatrix}$$

Casos C3a y C3b

$$\begin{pmatrix} \begin{pmatrix} [B_1]_{t_{1,1}} & [B_2]_{t_{1,1}} & [B_3]_{t_{1,1}} \\ [B_1]_{t_{2,1}} & [B_2]_{t_{2,1}} & [B_3]_{t_{2,1}} \\ \vdots & \vdots & \vdots \\ [B_1]_{t_{50,1}} & [B_2]_{t_{50,1}} & [B_3]_{t_{50,1}} \end{pmatrix} \\ \begin{pmatrix} [B_1]_{t_{1,2}} & [B_2]_{t_{1,2}} & [B_3]_{t_{1,2}} \\ [B_1]_{t_{2,2}} & [B_2]_{t_{2,2}} & [B_3]_{t_{2,2}} \\ \vdots & \vdots & \vdots \\ [B_1]_{t_{50,2}} & [B_2]_{t_{50,2}} & [B_3]_{t_{50,2}} \end{pmatrix} \\ \dots \dots \dots \\ \dots \dots \dots \\ \dots \dots \dots \\ \begin{pmatrix} [B_1]_{t_{1,45}} & [B_2]_{t_{1,45}} & [B_3]_{t_{1,45}} \\ [B_1]_{t_{2,45}} & [B_2]_{t_{2,45}} & [B_3]_{t_{2,45}} \\ \vdots & \vdots & \vdots \\ [B_1]_{t_{50,45}} & [B_2]_{t_{50,45}} & [B_3]_{t_{50,45}} \end{pmatrix} \end{pmatrix}$$

Casos C4a, C4b y C4c

C4a

$$\begin{pmatrix} [B_1]_{t_{1,1}} & [B_2]_{t_{1,1}} & [B_3]_{t_{1,1}} & [B_1]_{t_{1,2}} & [B_2]_{t_{1,2}} & [B_3]_{t_{1,2}} & \dots \dots \dots & [B_1]_{t_{1,45}} & [B_2]_{t_{1,45}} & [B_3]_{t_{1,45}} \\ [B_1]_{t_{2,1}} & [B_2]_{t_{2,1}} & [B_3]_{t_{2,1}} & [B_1]_{t_{2,2}} & [B_2]_{t_{2,2}} & [B_3]_{t_{2,2}} & \dots \dots \dots & [B_1]_{t_{2,45}} & [B_2]_{t_{2,45}} & [B_3]_{t_{2,45}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \dots \dots & \vdots & \vdots & \vdots \\ [B_1]_{t_{50,1}} & [B_2]_{t_{50,1}} & [B_3]_{t_{50,1}} & [B_1]_{t_{50,2}} & [B_2]_{t_{50,2}} & [B_3]_{t_{50,2}} & \dots \dots \dots & [B_1]_{t_{50,45}} & [B_2]_{t_{50,45}} & [B_3]_{t_{50,45}} \end{pmatrix}$$

C4b

$$\begin{pmatrix} A_{1,t_1,1}^{S_1} & A_{2,t_1,2}^{S_1} & A_{3,t_1,3}^{S_1} \\ A_{1,t_2,1}^{S_2} & A_{2,t_2,2}^{S_2} & A_{3,t_2,3}^{S_2} \\ \vdots & \vdots & \vdots \\ A_{1,t_{50},1}^{S_3} & A_{2,t_{50},2}^{S_3} & A_{3,t_{50},3}^{S_3} \end{pmatrix}$$

C4c

$$\begin{pmatrix} \alpha_{2,t_1,1} & \alpha_{2,t_1,2} & \dots & \alpha_{2,t_1,45} \\ \alpha_{2,t_2,1} & \alpha_{2,t_2,2} & \dots & \alpha_{2,t_2,45} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{2,t_{25},1} & \alpha_{2,t_{25},2} & \dots & \alpha_{2,t_{25},45} \end{pmatrix}$$

*Creyó en lo imposible y, precisamente por eso,
Venció batallas que todos considerábamos perdidas.
Ésa es la cualidad del guerrero:
entender que voluntad y coraje no son lo mismo.
Aleph*

VI. Influencia de distintas Variables y Respuesta de la Red Neuronal

Capítulo VI.

Influencia de distintas Variables y Respuesta de la Red Neuronal

El presente Capítulo está dedicado a la evaluación y análisis de la influencia de distintas variables en el proceso de *Entrenamiento* y *Predicción* de la Red Neuronal Artificial y la respuesta de la Red a estas influencias estudiadas desde un punto de vista general y cualitativo. El enfoque cuantitativo y específico, considerando los casos particulares del tratamiento ANN de los sistemas cinéticos que son de nuestro interés, constituye el contenido del Capítulo siguiente.

En el primer apartado de este Capítulo se exponen aspectos generales de la Arquitectura (número de capas), Configuración (número de neuronas o nodos de la/s capa/s Oculta/s), elementos matemáticos involucrados (matrices) y en general, las principales características estructurales de una Red Neuronal Artificial. En el siguiente apartado se analizan las acciones de las variables sobre el proceso de *Entrenamiento* o *Aprendizaje*, donde hemos considerado las distintas influencias sobre cada una de las capas que constituyen la Red Neuronal. De esta manera, las variables que inciden en la capa de Entrada (*Input layer*), como son, el número de curvas, número de puntos por curva, porcentaje de conversión de la reacción, valores de porcentajes relativos tanto de *Training*, *Validation* y *Testing* como la relación de $\epsilon_1/\epsilon_2/\epsilon_3$, constituyen el sub-apartado VI.1.2 y los sub-apartados VI.1.3 el caso de la/s capa/s oculta/s (*hidden layer*) y VI.1.4, el de la capa de salida (*output layer*).

VI.1.-Influencias de distintas variables en ANN

Antes de comentar el procedimiento que hemos seguido para la evaluación de la influencia que ciertas variables poseen en los resultados obtenidos tras la aplicación de ANN a sistemas cinéticos, vamos a compendiar en un único apartado y explicar, de una manera concisa y esquemática, la Arquitectura (número de capas), Configuración (número de neuronas o nodos de la/s capa/s Oculta/s), elementos matemáticos involucrados (matrices) y en general las principales características estructurales de una Red Neuronal Artificial.

VI.1.1.- Arquitectura de la Red Neuronal ANN. Capas de neuronas

Para la explicación, consideraremos los ejemplos reales del trabajo y realizaremos una exposición de modo general para todos los sistemas tratados sin centrarnos en ninguno en particular, ya que lo haremos posteriormente, en el capítulo de Resultados y Discusión.

VI.1.2.- Capa de Entrada (*Input Layer*)

Está configurada por n_d neuronas, que coincide con el número de datos cinéticos adquiridos a los distintos tiempos en cada curva cinética. La información de entrada a la red se realiza suministrándole 2 matrices, la matriz de *Inputs* y la matriz de *Targets*, que poseen la siguiente composición:

- Matriz de *Inputs* ($n_d \times n_c$): los elementos de sus filas corresponden a los n_d datos de entrada (*Inputs*) y son los valores de absorbancia total, de absorbancia individual, concentración de cada especie o fracción remanente individual, para cada curva cinética; los elementos de sus columnas, corresponden a los datos cinéticos de n_c curvas cinéticas distintas, generados sintéticamente a partir de los valores de los parámetros ordenados de acuerdo a lo que se establece en el Diseño de Experimentos (ED).

- Matriz de *Targets* ($n_p \times n_c$): los elementos de sus filas son los valores de los parámetros cinéticos y los elementos de sus columnas corresponden a los datos cinéticos de n_c curvas cinéticas distintas.

VI.1.3.- Capa/s Oculta/s (*Hidden Layer/s*)

Una vez que la Red ha distribuido la totalidad de los datos cinéticos en 3 bloques para efectuar los procesos de *Training* (TR) propiamente dicho, de *Validation* (VL) y *Testing* (TS), hemos de elegir “*a priori*” el número de capas y el número de nodos o neuronas que componen cada capa Oculta (*Hidden*) en orden a completar el proceso global de *Entrenamiento*. Tenemos que determinar el número óptimo de capas ocultas (1,2,3...) y el número óptimo de nodos o neuronas de cada una de esta/s capa/s ($n_1/n_2/n_3 \dots$), de acuerdo al procedimiento que posteriormente explicaremos.

VI.1.4.- Capa de Salida (*Output Layer*)

Está constituida por n_p neuronas y posee una matriz de *Outputs* ($n_p \times n_c$) donde se recogen las salidas de los parámetros de las n_c curvas, obtenidos por la ANN en cada una de las diferentes iteraciones (*Epochs*) tras la minimización matemática que realiza el algoritmo de optimización respecto a la matriz de *Targets* que se ha suministrado a la ANN en los datos de *Inputs*. Las matrices de *Outputs* y *Targets* poseen el mismo número de elementos y las mismas dimensiones ($n_p \times n_c$).

VI.2.- Influencia en el Proceso de *Entrenamiento o Aprendizaje*

El estudio de evaluación de la influencia que distintas variables ejercen en los resultados de aplicabilidad de ANN, lo hemos efectuado programándolo de acuerdo a las pautas que caracterizan al Diseño de Experimentos (ED). Es decir, hemos seleccionado un conjunto de variables que influyen en el comportamiento de las ANN (son los denominados *Factores*) y hemos evaluado las Respuestas individualizadas por parte de la Red Neuronal de las distintas variables, una vez que se han fijado o mantenido constantes los valores de las demás. Vamos a comentar a continuación de una manera esquemática, las características principales del procedimiento que hemos seguido para la evaluación de estas influencias. Hemos considerado un conjunto de variables (agrupadas de acuerdo a las capas en las que intervienen) que modifican e influyen en el proceso de *Entrenamiento* de la Red Neuronal (ANN) y que son las siguientes:

VI.2.1- Variables que intervienen en la Capa de Entrada (*Input Layer*):

Influencia en la matriz de *Inputs* de:

VI.2.1.1. - Número de datos de *Inputs* (n_d). Configuración de la capa de *Inputs*.

VI.2.1.2.- Número de curvas sintéticas (n_c) que se suministran a la Red ANN.

VI.2.1.3.- Variación en el porcentaje de reacción (conversión) en cada curva cinética.

VI.2.1.4.- Valores relativos de los porcentajes de distribución óptimos de los procesos. (*Training (TR) Validation (VL) Testing (TS)*, del Proceso Global de *Entrenamiento*).

VI.2.1.5.- Distintas relaciones de valores $\varepsilon_1/\varepsilon_2/\varepsilon_3$ en el caso de las Cinéticas Isotérmicas (Diseño Experimentos).

VI.2.2.- Variables que intervienen en la Capa/s Oculta/s (*Hidden Layer/s*):

VI.2.2.1.- Descripción del procedimiento para establecer el número óptimo de capas y el número óptimo de nodos o neuronas por capa/s.

VI.2.3.- Variables que intervienen en la Capa de Salida (*Output Layer*):

Matriz de *Outputs*.

VI.2.3.1.- Influencia de la aplicación de diversos Algoritmos de optimización que permite utilizar MATLAB.

Influencias VI.2.1.1, VI.2.1.2 y VI.2.1.3

Debido a las similitudes que presentan las influencias **VI.2.1.1**, **VI.2.1.2** y **VI.2.1.3** vamos a considerarlas de forma conjunta, aunque obviamente cada una de ellas hay que estudiarlas separadamente, esto es, fijando todas las variables excepto la que es objeto de estudio. Se trata de suministrar a la Red Neuronal en la matriz de *Inputs*, series de valores de absorbancias, concentraciones o fracciones remanentes que posean:

- distinto número de datos cinéticos (n_d) para el mismo intervalo de conversión y mismo número de curvas (n_c) (Influencia **VI.2.1.1**). Este valor, determina la Configuración o número de neuronas de la capa de Entrada (*Input Layer*).
- distinto número de curvas cinéticas (n_c) para el mismo intervalo de conversión y mismo número de datos cinéticos (n_d) (Influencia **VI.2.1.2**).
- distinto intervalo de conversión manteniendo constante el número de datos cinéticos y mismo número de curvas (Influencia **VI.2.1.3**).

El resultado obtenido de estas 3 variaciones individuales ha sido el esperado. Resulta trivial que a mayor número de datos por curva, mayor intervalo de conversión y mayor número de curvas, se obtengan mejores resultados de Regresión y Análisis de Residuales (MSE, Parámetros de las Rectas de Regresión, factor R, Distribución estadística de Residuales,... etc.). Es evidente que todos estos resultados se producen con un aumento considerable en el trabajo y tiempo de computación (*Epochs* o número de iteraciones). El criterio que debemos adoptar es aquel en que el punto deseado es un umbral que a partir de él sigue produciéndose una ligera mejora de resultados al aumentar el número de curvas, la amplitud del intervalo y el número de *Inputs*, aunque dicha mejora se produce en menor medida. Es decir, el punto óptimo es aquel en que la mejora de resultados ya no justifica el enorme aumento del trabajo y tiempo de computación y donde la mejora ya no es significativa. Por tanto, el término óptimo lo empleamos en el sentido de idóneo o apropiado.

Posteriormente, exponemos en detalle las magnitudes o valores óptimos que deben tener estas variables para los casos particulares que nos ocupan. Obtenemos los resultados a partir de la variación controlada de las variables consignadas en los apartados **VI.2.1.1**, **VI.2.1.2** y **VI.2.1.3**.

Influencia VI.2.1.4. Valores relativos de los porcentajes de distribución óptimos de los procesos *Training* (TR), *Validation* y (VL) *Testing* (TS).

En el proceso de *Entrenamiento* de una Red Neuronal, MATLAB exige que los datos sintéticos le sean suministrados previamente formando una base de curvas organizadas dentro de la matriz de *Inputs* y los parámetros que sirvieron para generarlos, en la matriz de *Targets*. Además, hay que suministrarle los valores de los porcentajes de curvas que deseemos sean utilizadas para realizar específicamente las tareas de *Training*, *Validation* y *Testing*. Nosotros le imponemos el valor del porcentaje de curvas del total de la base que queremos sean sometidas a las tareas de *Validation* y *Testing* y el programa informático elige de manera aleatoria un número de curvas que satisfaga el porcentaje que le hemos impuesto. Por tanto, hay 3 bloques o grupos de curvas: el conjunto completo de la base se utiliza para el proceso de *Entrenamiento* propiamente dicho, un subconjunto con un porcentaje impuesto por nosotros, que se somete a

Validación en cada una de las iteraciones (*epochs*) y otro porcentaje de curvas también impuesto por nosotros, que se utiliza para la tarea de *Testing*. Es muy importante señalar que en nuestro caso particular en el que aplicamos ANN a sistemas cinéticos, no hay dificultades con la elección de porcentajes pues carecemos de determinadas restricciones que en otros casos si existen. Los datos cinéticos que componen la base de curvas, los obtenemos mediante generación sintética de los mismos mediante la ejecución del correspondiente programa MATLAB que hemos diseñado y escrito al efecto. En otros ámbitos, esta base de curvas de datos de *Inputs*, corresponde a patrones conocidos que hay que obtener experimentalmente en el laboratorio. Esto puede representar un inconveniente insalvable e ineludible que restrinja o impida generar un colectivo de curvas patrón tan amplio que asegure 3 bloques de datos con un número suficiente cada uno para realizar todas las tareas para alcanzar los objetivos.

En todo caso, conviene tener presente algunos conceptos importantes que nos permitan elegir los mejores porcentajes y que vamos a adoptar como criterio para obtener los mejores resultados. El *Entrenamiento* es un proceso iterativo en el que la ANN va obteniendo nuevos valores de los parámetros deseados que ubica en la matriz de *Outputs* y que posteriormente mejora iterativamente mediante un proceso de optimización matemática por minimización de las denominadas señales o elementos de las matrices de *Outputs* y *Targets*. En cada iteración (*epoch*) o bien después de realizar un conjunto de ellas, MATLAB selecciona de manera aleatoria un número de curvas de acuerdo al porcentaje que le hemos ordenado que van a someterse a *Validación*. Para ello, calcula el valor del error MSE (*Mean Square Error*) entre las señales correspondientes a ambas matrices (*Outputs* y *Targets*). Tras almacenar el valor actual de MSE, repite el proceso completando la siguiente iteración (*epoch*) que compara con el de la iteración precedente. Si es inferior, lo almacena y continúa el proceso iterativo. Si es superior, continúa el proceso, salvo que se trate de la sexta vez consecutiva que esto sucede, en cuyo caso, finaliza el proceso iterativo. El programa computacional MATLAB, analiza la secuencia de valores de MSE de *Validation* tras varias iteraciones y selecciona el mínimo valor de MSE alcanzado en todo el proceso y es precisamente el que asigna como valor MSE del proceso global de *Entrenamiento*. Hay otros 2 criterios facultados para finalizar el proceso iterativo, como es el hecho de que se alcancen 1000 iteraciones o bien que el valor del módulo del vector gradiente sea inferior a 10^{-10} que convertiría en *infinitamente* lento e inacabable el avance al mínimo y por tanto, el final del Proceso global de *Entrenamiento*. Mientras esto no ocurra, el proceso de *Training* continúa ya que los valores de MSE de la base total de curvas de *Entrenamiento* siguen disminuyendo mientras no se satisfagan los criterios de finalización anteriormente comentados. A modo de resumen diremos que el proceso de *Validación*, es un mecanismo de control y monitorización del proceso de *Entrenamiento* de la Red Neuronal que garantiza su buen funcionamiento y sus buenos resultados.

El proceso de *Testing* es similar al de *Validación*. Selecciona un subconjunto de curvas de acuerdo al porcentaje que hayamos asignado previamente, comparando las denominadas señales o elementos de las matrices de *Outputs* y *Targets*, que posibilita la determinación del valor de MSE en cada unidad de tiempo (p.ej. cada segundo) y en la medida que ambas señales coincidan, el error MSE se aproximará asintóticamente a cero. La diferencia con la *Validación* es que la monitorización y control del proceso, se realiza en función del tiempo aunque podemos visualizar la evolución del *Testing* en una superposición de todas las tareas conjuntamente, cambiando la escala de tiempos por el número de *epochs*.

Es obvio que ha de existir coincidencia en la evolución de las curvas de *Testing* y *Validación*, es decir, han de transcurrir siempre muy próximas y en especial cuando se produzca el mínimo de MSE de la de *Validación*. Por tanto, éste debe ser un criterio válido para catalogar el proceso de *Entrenamiento* como correcto y satisfactorio.

Influencia VI.2.1.5. Influencia de las distintas relaciones de valores $\varepsilon_1/\varepsilon_2/\varepsilon_3$

Como es bien sabido, los datos cinéticos isotérmicos, cuyo tratamiento realizamos con la técnica de ANN, dependen directamente de los valores de las concentraciones y de los coeficientes de absorción molar de cada una de las especies. Deseamos evaluar la respuesta de la ANN y la influencia que posee la variación de la magnitud de los valores (absolutos y relativos) de los coeficientes de absorción molar en el *Entrenamiento* y en la determinación de los parámetros cinéticos de los distintos sistemas. Para su estudio, hay que garantizar la variación continua e individualizada de los coeficientes de absorción molar y considerar todas las posibilidades de variación para que cualquier experiencia cinética realizada en el laboratorio, esté incluida en el espacio que hemos estudiado. La mejor manera de conseguir esto, es organizar las variaciones individuales de todos los *Niveles* entre sus valores extremos.

El análisis de los resultados obtenidos nos han permitido evaluar las correlaciones entre los resultados y las variaciones sistemáticas de los relaciones de los coeficientes de absorción molar que permiten evaluar el tipo y magnitud de la influencia. Hemos de hacer una consideración importante relativa a los puntos en los que se cumple $\varepsilon_1 = \varepsilon_2$ o bien $\varepsilon_1 = \varepsilon_2 = \varepsilon_3$. Desde el punto de vista de la Cinética Química isotérmica experimental clásica, estos puntos son ineficaces y su utilización experimental absurda, puesto que corresponden a valores constantes de absorbancia total (A_T) con la variación de tiempo. No hay variación de A_T con el progreso de la reacción y en consecuencia no son válidos para realizar un estudio cinético clásico de la reacción. Corresponden a puntos con valores de ε extremos, que delimitan el *Dominio Experimental* al que pertenecen las posibles cinéticas que se realizan en el laboratorio y por tanto no se pueden suprimir en el ED. Con este tipo de datos, el comportamiento computacional ANN que hemos observado es más bien dispar, esto es, algunas veces se

obtienen resultados inaceptables al tratarse de valores extremos. Como su intervención en el *Entrenamiento* de la Red Neuronal es imprescindible y no pueden eliminarse, en caso de encontrar malos resultados, hay que modificar las matrices de *Input*, sustituyendo los valores de absorbancias totales por absorbancias individuales disponiéndolos en forma secuencial, que permita realizar un *Entrenamiento* con éxito.

Influencia VI.2.2. Influencia del número de capas Ocultas (Estructura) y de Nodos por cada capa (Configuración). Descripción del Procedimiento

El establecimiento de la Arquitectura de la Red Neuronal (ANN) implica la realización de un estudio específico para conocer su Estructura (número óptimo de capas ocultas) y la Configuración (número de nodos o neuronas) de cada una de ellas. Para ello, puede desarrollarse un procedimiento que supone la aplicación de un método del tipo “*Prueba y Error*”. Este método consiste en elegir inicialmente y fijar el número de capas ocultas (p. ej., Estructura inicial de una capa) constituidas por un determinado número de nodos (1,2,3,4,5,6...etc.) obteniendo Configuraciones distintas que posteriormente se someten al proceso de *Entrenamiento* tras haber fijado o mantenido constantes las acciones de las demás variables que influyen en el *Aprendizaje* de la ANN. Se realiza el *Entrenamiento* de todas y cada una de las diferentes Configuraciones para cada Estructura invariable, obteniendo los valores de los parámetros de Regresión y Análisis de Residuales del proceso global de *Entrenamiento* y los de los grupos de *TR*, *VL* y *TS*. Esto es, los valores correspondientes de MSE mínimo de *Validación* de cada Configuración, los parámetros de las Rectas de Regresión y sus valores de R; distribución de Residuales; valores de μ (“*momentum constant*”); *Performance*; número de *epoch* en el que la *Validación* fracasa,...etc. El procedimiento se repite para las demás Configuraciones correspondientes a Estructuras constituidas por 2 capas, 3 capas,...etc. y una vez obtenidos los resultados para todas las Configuraciones con distinto número de capas ocultas, se realiza para cada Estructura una representación gráfica, de los valores obtenidos de MSE mínimos, frente a los valores del número de nodos o neuronas que componen cada capa oculta. Obtenemos figuras que presentan curvas con perfiles descendentes, observando que cada curva posee 2 tramos bien diferenciados: un primer tramo (para valores menores del número de nodos) con pendiente negativa muy acusada y un segundo tramo también con pendiente negativa mucho menor, con perfil de “meseta descendente”. Si hacemos una suavización o *smoothing* de la ubicación cartesiana de los puntos de ambos tramos y prolongamos sus líneas de tendencia, el punto de corte nos proporciona en abscisas el valor del número óptimo de neuronas o nodos que debe asignarse a la configuración de la Capa oculta. La explicación de este criterio de asignación del óptimo es sencilla y bastante intuitiva. Se trata de encontrar el número umbral de nodos a partir del cual, el aumento del número de nodos produce

una suave y sostenida disminución de los valores del MSE pero en una medida mucho menor que la que se produce en el tramo anterior al número umbral. Es evidente que al aumentar el número de nodos el valor de MSE disminuye pero esta leve mejora no justifica el extraordinario aumento del trabajo y tiempo de computación. Es decir, se trata de encontrar el punto a partir del cual, los valores de MSE siguen disminuyendo pero la mejora ya no resulta significativa. En algunas ocasiones el punto de corte coincide con un mínimo de la curva más o menos pronunciado lo que facilita la elección.

Respecto a esta metodología tenemos que hacer una puntualización importante relativa al significado del término óptimo (“*el mejor*” o “*inmejorable*”), usado por multitud de investigadores en el campo de ANN, ya que puede resultar, cuando menos impreciso. La Arquitectura encontrada por este procedimiento produce los mejores resultados del proceso de *Entrenamiento* para un particular conjunto de valores de las variables que intervienen en el mismo. Es probable que exista otro conjunto distinto de valores de las variables con el que se obtengan similares o mejores resultados y para determinarlo, habría que probar para infinidad de conjuntos de valores cada uno con una Arquitectura distinta que supondría entrar en un “círculo vicioso” que convertiría obviamente el proceso en utópico e irrealizable. La estrategia acertada consiste en probar con unos cuantos conjuntos de valores intermedios (no extremos) de las variables y seleccionar aquel que proporcione los resultados idóneos o más adecuados. En todo caso, incluso la arquitectura que proporciona el valor elegido del número de nodos umbral, tampoco corresponde al óptimo, pues un mayor número de nodos produce similares o mejores resultados. Podemos concluir diciendo que no pretendemos encontrar la Arquitectura que produzca los mejores resultados (óptima) cuya calidad no pueda ser superada por otra arquitectura diferente. Lo que deseamos es que los parámetros de ajuste del *Entrenamiento* (MSE, Regresión y Coeficiente R,...etc.) garanticen satisfactorios resultados de la *Predicción* de parámetros cinéticos desconocidos de curvas experimentales obtenidas en el laboratorio, que es el objetivo último que pretendemos alcanzar. En este trabajo hemos utilizado el término “óptimo” pero para evitar ambigüedades, está afectado por el matiz que acabamos de comentar.

En la Figura VI.1.1 mostramos un ejemplo de este tipo de curvas correspondiente al sistema cinético isotérmico de reacciones consecutivas (base de 45 curvas con 150 datos cada una, en concentraciones de *Input* y ordenación secuencial) donde puede observarse que la configuración óptima para la Estructura de una capa oculta corresponde a 8 nodos. Hemos considerado la Estructura de una única capa oculta lo cual no quiere decir que con las Estructuras de 2, 3,..etc. capas se obtengan mejores resultados. Hay que recordar que en este Capítulo únicamente pretendemos conocer el procedimiento para evaluar la Respuesta de la ANN y la influencia de las diferentes y Configuración/es de una misma Estructura. Posteriormente, exponemos los datos y resultados cuantitativos obtenidos para distintas Estructuras de capas ocultas y diferentes Configuraciones, donde aplicamos este procedimiento

para la determinación de la Estructura (número de capa/s oculta/s) y de la/s Configuración/es óptimas de las Redes Neuronales Artificiales para el tratamiento de sistemas cinéticos reales.

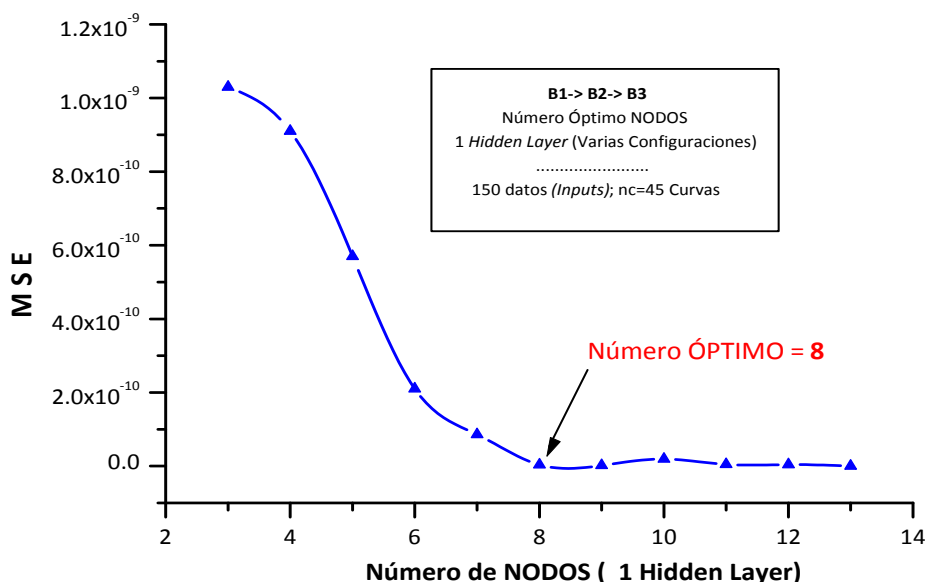


Figura VI.1.1: Ejemplo correspondiente al sistema cinético isotérmico de reacciones consecutivas donde puede observarse que la configuración óptima para la Estructura de una capa oculta corresponde a 8 nodos.

Influencia VI.2.3. Variables que intervienen en la Capa de Salida (*Output Layer*):

En la capa de salida se genera la matriz de *Outputs*. Se trata de una matriz cuyos elementos son calculados por la Red Neuronal y no hay posibilidades de estudio de influencias que puedan modificarla. La Configuración de dicha capa (número de neuronas de salida) no puede modificarse ya que viene determinada por el número de parámetros cinéticos a *Predecir*. En cuanto a la incidencia de la acción de distintas variables, el factor más importante afecta a la optimización de los elementos de la matriz de *Outputs* frente a los de la matriz de *Targets* en cada una de las iteraciones (*Epochs*) tras aplicar un algoritmo de optimización. MATLAB ofrece la posibilidad de aplicación de una serie de algoritmos distintos. Hemos realizado un estudio comparativo de todas las posibilidades que ofrece MATLAB evaluando los resultados que se obtienen con cada uno de ellos al aplicarlos a los sistemas cinéticos objeto de nuestro estudio, en orden a poder elegir el que mejor resultados proporcione y que consideramos en el Apartado siguiente.

Influencia VI.2.3.1. Algoritmo de Optimización en el proceso de *Entrenamiento*

La Red Neuronal Artificial desarrolla la capacidad de *Aprendizaje* mediante el *Entrenamiento* de la misma. El método por el que se realiza el *Entrenamiento* es de gran importancia y para ello existen diferentes procedimientos tal y como hemos visto con anterioridad. A lo largo del desarrollo del trabajo que recoge esta Memoria hemos utilizado el algoritmo *Backpropagation*, un método de *Entrenamiento* de redes multicapa que utiliza las matrices de entrada (*Input*) y sus correspondientes matrices de *Targets* para *entrenar* la Red, mediante el ajuste de los *Outputs* de manera que disminuya progresivamente la diferencia entre la salida deseada y la respuesta de la Red. El algoritmo *Backpropagation* se refiere a la forma en que se determinan los cálculos de las modificaciones de los *pesos* de las conexiones que empiezan por la capa de salida (*Outputs*) y continúan "hacia atrás", a través de todas las capas de la Red Neuronal hasta la capa de entrada (*Input*). El ajuste de los *pesos* se puede llevar a cabo mediante diferentes algoritmos. El método *Backpropagation* básico utiliza un algoritmo de Gradiente Descendente en el que los *pesos* de *Outputs* se mueven en la dirección del gradiente negativo de máximo descenso (*Steepest Descent Method*) (SDM). Existe una serie de variaciones sobre el algoritmo básico que emplean otras técnicas de optimización estándar, como son los métodos de Gradientes Conjugados o los métodos de Newton.

El diseño y aplicación de Redes Neuronales Artificiales que proporciona MATLAB ⁽⁶⁴⁾, permite la utilización de diversas técnicas de optimización matemática basadas en distintos algoritmos de minimización. La elección de una técnica u otra puede afectar al proceso de *Entrenamiento* de la red y es por este motivo por el que hemos analizado la aplicación de diversos algoritmos:

a) *Steepest Descent*

El método más sencillo de aprendizaje *Backpropagation*, es aquel en el que los *Outputs* se acomodan a la dirección en la que la función *MSE* (*Error Cuadrático Medio*) disminuye con mayor rapidez, *Steepest Descent* siguiendo la dirección del gradiente negativo. El algoritmo *Steepest Descent* se puede implementar de dos formas: el modo "incremental" y el modo por "lotes" (*batch*). En el modo "incremental", se calcula el gradiente y se actualizan los *Outputs* después de que se aplique cada entrada a la red. En el modo por "lotes", todas las entradas (*Inputs*) se aplican a la Red antes de que se actualicen las ponderaciones. Esta opción es la empleada por MATLAB y los algoritmos utilizados son:

- *Batch Gradient Descent*.
- *Gradient Descent with Momentum*.

Estos algoritmos son extraordinariamente lentos pero bastante seguros en cuanto al vector de movimiento ya que siempre sigue la dirección de descenso al mínimo.

b) Técnicas rápidas de *Entrenamiento (Faster Training)*

Los algoritmos incluidos en el primer bloque son muy lentos por lo que MATLAB proporciona otros más rápidos, que se dividen en dos categorías: aquellos que utilizan técnicas heurísticas y los que utilizan técnicas de optimización numérica.

b1) Técnicas heurísticas: MATLAB proporciona dos métodos basados en técnicas heurísticas:

- *Variable Learning Rate.*
- *Resilient Backpropagation.*

b2) Técnicas de optimización numérica: los métodos numéricos utilizados por MATLAB se dividen a su vez en dos grupos:

- **Algoritmos de gradiente conjugado:** el algoritmo *backpropagation* básico (*Steepest Descent*) ajusta los *pesos* en la dirección en la que la función *MSE* disminuye más rápidamente. La función *MSE* puede disminuir rápidamente pero no necesariamente produce una convergencia rápida. En los algoritmos de gradiente conjugado se realiza una búsqueda a lo largo de las direcciones conjugadas, que produce por lo general una convergencia más rápida. Dentro de este grupo MATLAB permite la aplicación de los siguientes algoritmos:

- *Fletcher-Reeves Update.*
- *Polak-Ribière Update.*
- *Powell-Beale Restarts.*
- *Scaled Conjugate Gradient.*

- **Algoritmos *Quasi-Newton*:** estos métodos son una alternativa a los métodos de gradiente conjugado y conducen a una optimización más rápida. El método de Newton es un método iterativo, que utiliza como paso básico para la optimización la ecuación:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \quad \text{[VI.1.1]}$$

donde \mathbf{H}_k^{-1} es la matriz Hessiana (segundas derivadas) de la función *MSE* y \mathbf{g}_k el vector gradiente.

Los métodos de Newton normalmente convergen más rápidamente que los de Gradiente Conjugado, pero el cálculo de la matriz Hessiana es complejo y requiere mucho tiempo de computación, sobre todo en el caso de las denominadas redes *feedforward*. Los métodos *Quasi-Newton* no necesitan del cálculo de las segundas derivadas puesto que en cada iteración (*Epoch*) se calcula una matriz Hessiana aproximada y el proceso se simplifica considerablemente. Los algoritmos *Quasi-Newton* proporcionados por MATLAB son los siguientes;

- *BFGS Algorithm.*
- *One Step Secant Algorithm.*
- *Algoritmo de Levenberg-Marquardt.*

Al igual que los métodos Quasi-Newton el algoritmo Levenberg-Marquardt determina de forma aproximada la matriz Hessiana y el vector gradiente de la función MSE de la siguiente manera:

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} \quad \text{[VI.1.2]}$$

$$\mathbf{g} = \mathbf{J}^T \mathbf{e} \quad \text{[VI.1.3]}$$

Donde \mathbf{e} es un vector de errores y \mathbf{J} es la matriz Jacobiana que contiene las primeras derivadas y que resulta más sencilla de calcular computacionalmente. A partir de la matriz Hessiana aproximada, el algoritmo Levenberg-Marquard realiza la optimización de manera iterativa mediante la ecuación:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e} \quad \text{[VI.1.4]}$$

En el Capítulo VII, mostramos los resultados obtenidos tras la aplicación de todos estos algoritmos en el proceso de *Entrenamiento* de la Red Neuronal que corresponde tratamiento ANN de los sistemas cinéticos estudiados. Mantenemos fijas todas las posibles variables que intervienen en el *Aprendizaje* o *Entrenamiento* de Red Neuronal, excepto el algoritmo *Backpropagation*, que aplicamos todos los que proporciona MATLAB⁽⁶⁴⁾, de forma individual (uno a uno). La realización de un estudio comparativo entre los resultados que proporcionan, nos permite elegir y utilizar posteriormente en todas las experiencias computacionales, el algoritmo óptimo, idóneo o más adecuado en cada caso.

*Aprovecha todas las oportunidades de tu vida porque,
cuando pasan, tardan mucho tiempo en volver.
La Quinta Montaña*

VII. Resultados y Discusión

Capítulo VII.

Resultados y Discusión

El objetivo último del trabajo que se recoge la presente Memoria es el diseño y desarrollo de un nuevo Algoritmo Híbrido (AH) que permita la determinación de Parámetros Termodinámicos de Activación (PTA) y las constantes individuales de velocidad (k_r) de diversos mecanismos de reacción, además de la discriminación entre posibles mecanismos responsables del transcurso de la reacción. El AH propuesto está formado por la combinación de la metodología que utiliza Redes Neuronales Artificiales (ANN) y el algoritmo de Optimización Matemática AGDC. En una primera etapa hemos utilizado la metodología ANN para determinar los valores de los parámetros (PTA, k_r) y posteriormente se ha aplicado el algoritmo AGDC para mejorar los valores de dichos parámetros obtenidos en el proceso anterior.

El presente Capítulo está estructurado de la siguiente manera:

- Teniendo en cuenta las características de las diferentes metodologías que constituyen el AH (ANN-AGDC), en un primer apartado se exponen de manera concisa diferentes aspectos generales de ambas metodologías. A lo largo de este apartado se explica el procedimiento que se ha seguido para la aplicación de ambas metodologías además de los diferentes aspectos que influyen en los resultados obtenidos.
- Una vez expuestos los aspectos generales acerca de la aplicación de ANN y AGDC, se exponen los resultados obtenidos para la determinación de los Parámetros Termodinámicos de Activación (PTA) y las constantes individuales de velocidad (k_r) mediante la aplicación del AH en los diferentes mecanismos de reacción.
- Por último se exponen los resultados obtenidos cuando se aplica ANN y AGDC para la discriminación de mecanismos de reacción, es decir establecer el modelo más probable al que los datos cinéticos experimentales pertenecen.

VII.1.- Consideraciones Generales del Algoritmo Híbrido

A lo largo de este apartado se exponen una serie de consideraciones generales acerca de la aplicación del AH (ANN-AGDC). La estructura de este apartado es coincidente con la exposición de los resultados obtenidos en cada uno de los sistemas estudiados, en primer lugar se explican aspectos relativos a la aplicación de ANN y a continuación las características del proceso de Optimización Matemática mediante AGDC.

VII.1.1.- Redes Neuronales Artificiales (ANN)

Consideramos en primer lugar todos los aspectos relacionados con la aplicación de la metodología ANN para la determinación de Parámetros Termodinámicos de Activación y constantes individuales de velocidad. En primer lugar analizamos todas las consideraciones que son necesarias tener en cuenta para realizar los diferentes Diseños de Experimentos, a continuación se considera el Proceso de *Entrenamiento* de la ANN analizando los valores óptimos de la Arquitectura y de sus variables. Por último se describe el proceso genérico de *Predicción* de la Red Neuronal.

VII.1.1.1.- Diseños de Experimentos

Hemos evaluado la influencia del número de curvas en el proceso de *Entrenamiento* de la Red para determinar de esta manera el número óptimo (de nuevo, matizado en el sentido de idoneidad) de curvas que han de componer la base de *Entrenamiento*. Para alcanzar este objetivo se han llevado a cabo distintas experiencias computacionales sintéticas en las que hemos modificado el número de curvas disponibles para el *Entrenamiento*, manteniendo fijas todas las demás variables. Hemos sistematizado el número de curvas generadas sintéticamente de acuerdo a los pares o ternas de valores de las constantes de velocidad para Cinéticas Isotérmicas y en función de los Parámetros Termodinámicos de Activación en Cinéticas No Isotérmicas, organizadas según un Diseño de Experimentos (ED) del tipo *Central Star Composite Experimental Design*, (CSCED). Hemos generado conjuntos de 59, 45, 41, 33, 29 y 25 curvas cada uno de ellas.

Hemos evaluado la influencia del número de curvas en 3 supuestos: *a)* para una Arquitectura con una capa Oculta (*Hidden*) considerando una Configuración para esta capa constituida por distintos nodos o neuronas, *b)* para una Arquitectura de dos capas (*Hidden*) con Configuraciones de ambas capas ocultas de diferentes nodos o neuronas y *c)* para una Arquitectura con tres capas (*Hidden*) con Configuraciones variadas de todas las capas ocultas. En la mayoría de las situaciones, los mejores resultados (menor valor de MSE de *Validación*) se obtienen en el caso de que el *Entrenamiento* se realice con 59 curvas (ED en 3D) y 45 curvas (ED en 2D), puesto que los valores de todos los parámetros que nos proporciona el análisis de los resultados del proceso de *Entrenamiento* son sensiblemente mejores, tal como cabía esperar.

VII.1.1.2.- *Entrenamiento de la ANN. Valores Óptimos de la Arquitectura y de sus Variables*

En todos los tratamientos en los que intervienen las técnicas basadas en Redes Neuronales Artificiales, resulta crucial establecer con seguridad la Arquitectura óptima de la Red con referencia a las 3 capas que la constituyen, esto es, capa de Entrada (*Input layer*), capa/s Oculta/s (*Hidden layer/s*) y capa de Salida (*Output layer*). Es necesario determinar “*a priori*” la Estructura y la Topología de la Red Neuronal para conocer el número óptimo de capas, la Estructura de cada una y sus posibles Configuraciones (número de neuronas o nodos), que proporcionan los mejores resultados en el proceso de *Entrenamiento* de la ANN.

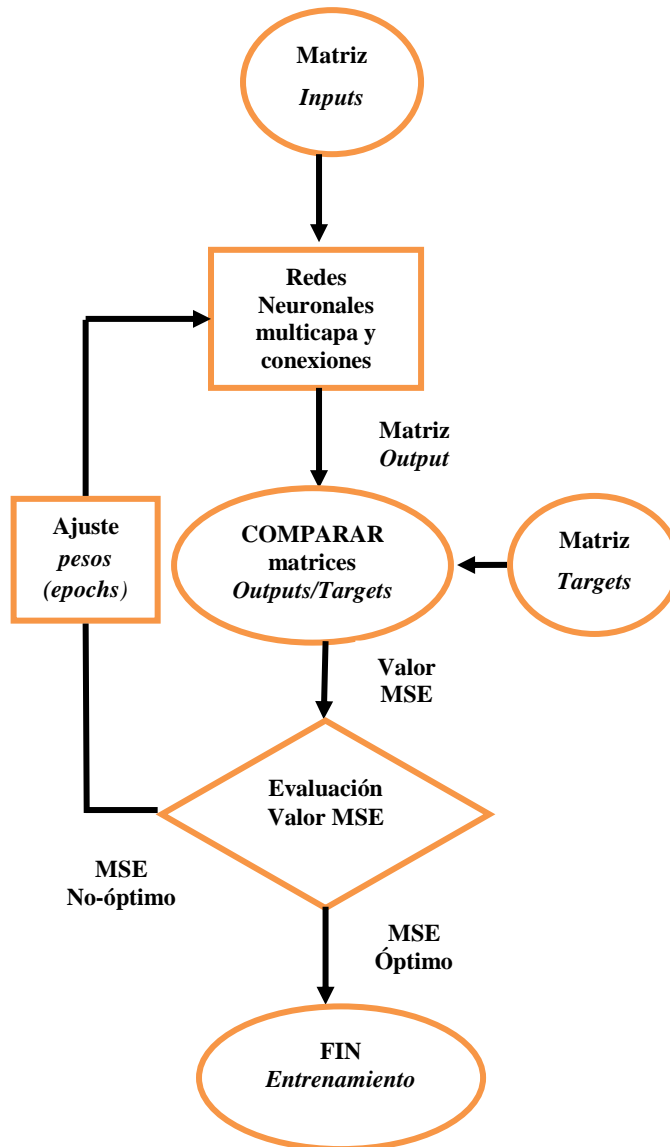
La capa de Salida (*Output layer*) no tiene prácticamente opciones de elección pues viene establecida por el número de parámetros que intervienen en el sistema cinético objeto de estudio. La capa de Entrada (*Input layer*), permite libertad de elección al tener que decidir el número óptimo de datos de entrada que proporcione los mejores resultados tras realizar el Análisis de Residuales del proceso de *Entrenamiento*. Este valor óptimo constituye una de las variables importantes cuya influencia cuantitativa hay que determinar “*a priori*”. Una vez que hemos determinado las características de las 2 capas anteriores, es necesario conocer el número óptimo de capas Ocultas y la Configuración óptima (número de neuronas o nodos) de las mismas que proporcionan los mejores resultados del *Entrenamiento* de la Red Neuronal. El procedimiento, que ya hemos explicado en detalle en el apartado de Influencias VI.2.2 ilustrado con la Figura VI.1.1, consiste en aplicar una técnica de “*Prueba y Error*” sometiendo a *Entrenamiento* o *Aprendizaje* la ANN en la que vamos modificando sistemáticamente la Configuración de la/a capa/s Oculta/s manteniendo constante el número de éstas (Estructura). Con todos los datos obtenidos, el perfil de la representación de los valores de MSE frente al número de nodos para cada Estructura o número de capas constante, proporciona la Configuración ideal óptima. Conocidas las Estructuras y Configuraciones de las 3 capas que producen los mejores resultados del *Entrenamiento* de la Red Neuronal, tendremos establecida la Arquitectura óptima de dicha ANN. Reiteramos una vez más que el término “óptimo” lo aplicamos con un cierto matiz, con el que deseamos significar que la Arquitectura óptima encontrada es para un conjunto particular de valores de las variables que intervienen en el *Entrenamiento*. Además, utilizamos la estrategia de probar con varios conjuntos de valores intermedios (no extremos) de las variables y aceptamos el que proporcione los mejores resultados. Probar para muchos conjuntos de valores en orden a encontrar la Arquitectura óptima, teniendo en cuenta que cada uno posee una Arquitectura diferente, sería entrar en un círculo vicioso que supondría un procedimiento utópico e irrealizable.

Una vez establecida la Arquitectura y Topología óptima de la Red Neuronal, tenemos que determinar de una forma cuantitativa y rigurosa, la influencia individual cuantitativa, de cada una de las diferentes variables y factores que afectan al proceso de *Entrenamiento* de la

Red Neuronal que aseguren los mejores resultados en el proceso de *Predicción*. Estos son: el número de curvas que componen la base de *Entrenamiento*; el número de puntos que componen cada una de estas curvas (*Inputs*); la distribución del porcentaje de curvas utilizadas para *Training*, *Validation* y *Testing*; el algoritmo de optimización de *Backpropagation* utilizado y las distintas relaciones de valores entre los coeficientes de absorción molar de las especies reaccionantes en las cinéticas isotérmicas.

El proceso descrito anteriormente es llamado *Entrenamiento* de Redes Neuronales multicapa y constituye un método iterativo donde después de cada iteración (*epochs*), los valores de los parámetros calculados están agrupados en la matriz de *Output* (b_{ij}^{output}) y son comparados con los correspondientes en la matriz de *Target* como se muestra en el siguiente esquema (Esquema VII.1.1) que resulta muy ilustrativo.

Esquema VII.1.1: Proceso de *Entrenamiento* de Redes Neuronales Artificiales Multicapa.



Los valores de MSE, expresados en valor absoluto, de acuerdo a las diferencias cuadráticas, son calculados de acuerdo a la expresión:

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{Outputs} - b_{ij}^{Targets})^2}{n_p \cdot n_c} \right)^{1/2} \quad [\text{VII.1.1}]$$

donde n_c es el número de curvas, n_p es el número de parámetros y $n_p \cdot n_c$ son las dimensiones de ambas matrices (matriz de *Outputs* y matriz de *Targets*).

Durante el proceso de *Entrenamiento*, los valores de los *pesos* se modifican utilizando algoritmos de optimización matemática adecuados con el fin de minimizar los valores de MSE calculados en cada iteración (*epoch*). En el presente trabajo, el algoritmo que hemos usado ha sido *Backpropagation*. El proceso iterativo concluye cuando se alcanza el valor mínimo de MSE, después de lo cual puede considerarse que el proceso ha terminado.

El *Entrenamiento* de la Red se completa con los procesos de *Validation* y *Testing*. Estos son dos procesos de control y verificación de la minimización del método iterativo entre las matrices de *Outputs* y *Targets*. Entre las diferentes curvas que comprenden la matriz de *Inputs*, se escoge un porcentaje del total, establecido previamente (5%, 10%, ...), lo que da lugar a una “*sub-matriz*” de curvas *Inputs* que están sometidas a un proceso iterativo hasta que el valor mínimo de MSE es alcanzado. Por tanto, es posible verificar la validez del proceso de *Entrenamiento*, garantizando que es convergente, y que tiene una finalización adecuada, ya que no existe *overfitting* y tampoco *overtraining*. El proceso de *Validation* se completa cuando el número de iteraciones consecutivas es mayor o igual a 6 manteniendo el valor de MSE constante o con una ligera tendencia a aumentar. El proceso de *Testing* es similar al de *Validation*, exceptuando que el control del proceso y la monitorización se realiza en función del tiempo aunque el usuario puede visualizar la tarea del *Testing* cambiando la escala de tiempos por el número de iteraciones (*epochs*).

VII.1.1.3.- Predicción de la Red Neuronal Óptima

El proceso genérico de *Predicción* de la Red Neuronal probablemente constituye el proceso más trascendente, puesto que en él se adopta la decisión definitiva sobre las curvas cinéticas para determinar los parámetros desconocidos. Lo hemos estructurado dividiéndolo en 2 apartados, el primero dedicado a la *Predicción* de curvas cinéticas con datos sintéticos a los que le hemos impuesto un error al azar (*noise*) del orden del experimental y el segundo con datos experimentales obtenidos en el laboratorio. Parece lógico evaluar en primer lugar la acción de las variables sobre los resultados de la *Predicción* de la ANN, aprovechando la ventaja que supone el conocimiento de los parámetros de entrada (*Targets*) utilizados para generar los datos. Así, evaluamos cuantitativamente la influencia en los resultados de la *Predicción*, las siguientes variables: nivel o valor del error de *noise* en los datos, número de

curvas a *Predecir*, número mínimo de especies necesarias y relación entre los coeficientes de absorción molar de las especies en el caso de cinéticas isotérmicas. La última etapa de la aplicación de las técnicas ANN, consiste en computar datos experimentales obtenidos en el laboratorio. En el presente trabajo hemos utilizado datos experimentales cinéticos existentes en bibliografía dotados de probada garantía y fiabilidad. Para el tratamiento ANN, hemos utilizado dos procedimientos: en algunas ocasiones los datos experimentales cinéticos son accesibles directamente, ya que los autores los han publicado explícitamente en el artículo de la propia revista científica especializada o en las correspondientes secciones de “Bases de datos”. En otras ocasiones esto no es posible y hemos tenido que generarlos con los datos y parámetros que proporcionan los autores (concentraciones iniciales, constantes cinéticas, coeficientes de absorción molar, parámetros de activación, etc.) así como las condiciones de reacción (temperatura, fuerza iónica, pH,..etc.).

Hemos incluido como apartado final a modo de resumen y debido a su importancia, un aclaratorio en el que compendiamos los distintos conceptos de las diferentes acepciones de las Desviaciones Estándar y Errores que hemos utilizado en ANN para expresar las precisiones de los resultados, dada la peculiaridad y similitud conceptual del tratamiento que se lleva a cabo en esta técnica computacional.

1.- Error MSE o Error Cuadrático Medio.

Viene definido por la ecuación [VII.1.1] y corresponde a la optimización entre los elementos numéricos de las matrices de *Outputs* y *Targets* ($b_{i,j}$) que se calcula al final de cada *Epoch* en el proceso iterativo de *Entrenamiento* de la ANN.

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{Outputs} - b_{ij}^{Targets})^2}{n_p \cdot n_c} \right)^{1/2} \quad [\text{VII.1.1}]$$

donde n_c es el número de curvas, n_p es el número de parámetros por curva, b_{ij} representa los parámetros cinéticos a determinar.

2.- Error Cuadrático Medio de los datos de *Input* o Error general del Ajuste.

Se define mediante la ecuación [VII.1.2], donde b_{ij}^{exp} es el valor del parámetro del dato de la matriz de *Inputs* y b_{ij}^{calc} es el valor del parámetro recalculado con los valores obtenidos determinados en el proceso de *Predicción* ([Y]), n_d es el número total de datos de la matriz de *Inputs* y n_c es el número de curvas. La desviación Estándar (SD) se calcula en la realización del Análisis de Residuales para determinar la Bondad del ajuste (*Residual Analysis* y *Goodness of the Fit*).

$$SD = \left(\frac{\sum_{i=1}^{n_d} \sum_{j=1}^{n_c} (b_{ij}^{exp} - b_{ij}^{calc})^2}{n_d \cdot n_c} \right)^{1/2} \quad [\text{VII.1.2}]$$

3.- Desviación Estándar o Error individual de cada una de las k cinéticas.

Únicamente puede calcularse cuando se computan datos sintéticos, ya que se conocen los valores de los parámetros que sirvieron para generar los datos sintéticos. Cuando se computan datos experimentales, no es posible calcular el error de cada parámetro en el proceso de *Predicción* al carecer de fórmulas matemáticas que permitan calcular la propagación del error correlacionando dicho error individual de los parámetros con el error general del ajuste [VII.1.2]. Sin embargo, en el caso de las técnicas de Regresión estas expresiones matemáticas existen. Tenemos que conformarnos con asignar exclusivamente un orden de magnitud al error individual de los parámetros cinéticos (p. ej. en la 5ª cifra decimal) cuando se conoce la relación existente entre el error de *noise* y el error individual de los parámetros calculado con datos sintéticos es conocida.

VII.1.2.- AGDC

Tal y como hemos señalado reiteradamente el AH propuesto está formado por la combinación de la metodología que utiliza Redes Neuronales Artificiales (ANN) y el algoritmo de Optimización Matemática AGDC. El procedimiento de aplicación del AH consiste en utilizar en primer lugar la metodología ANN para determinar los valores de los parámetros (PTA, k_r) siguiendo las diferentes etapas indicadas anteriormente, posteriormente se aplica el algoritmo AGDC para mejorar los valores de dichos parámetros. Nos centramos ahora en la aplicación del algoritmo AGDC que requiere llevar a cabo las siguientes etapas:

- Asignación de unos valores a los parámetros que se van a determinar (estimaciones iniciales), estos valores se optimizan posteriormente mediante un proceso iterativo. En nuestro caso las estimaciones iniciales de los parámetros son los valores obtenidos mediante la aplicación de la metodología ANN.
- Cálculo del valor de la propiedad del sistema utilizada en su estudio (concentración, absorbancia, fracción remanente α, \dots) a partir de estimaciones iniciales de los parámetros.
- Cálculo de la función *SQD* a partir de los valores experimentales y calculados de la propiedad con la que se monitoriza el sistema en estudio. La forma explícita de la función suma de desviaciones cuadráticas (SQD) no se conoce pero su valor en cada punto puede determinarse a partir de la ecuación [III.3.4]:

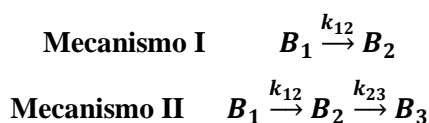
$$SQD = \sum_{i=1}^{N_d} (f_i(X_1, X_2, \dots, X_p, t_i) - (y_i)_E)^2$$

donde f_i representa la función analítica que proporciona el valor teórico de la propiedad (concentración, absorbancia, fracción remanente α, \dots), X_1, X_2, \dots, X_p

$(k_r; A_r, E_r; \Delta H_r^\ddagger, \Delta S_r^\ddagger)$ son los parámetros que se pretenden determinar y $(y_i)_E$ el valor experimental de la propiedad obtenida a los distintos valores de tiempo.

- Proceso de optimización matemática (minimización de la función SQD) mediante el algoritmo AGDC, para lo cual es necesario realizar una serie de cálculos numéricos:
 - Establecimiento del sistema de ecuaciones diferenciales de velocidad.
 - Resolución numérica de ecuaciones diferenciales.
 - Cálculo de las derivadas numéricas.
 - Determinación del vector gradiente y la matriz Hessiana (\mathbf{g} y \mathbf{H}).
 - Inversión de la matriz Hessiana (\mathbf{H}^{-1}).
- Una vez optimizados los parámetros se calculan los errores de los mismos y se lleva a cabo el análisis estadístico de los residuales.

El AH, y por tanto el algoritmo de optimización matemática AGDC ha sido empleado para la determinación de Parámetros Termodinámicos de Activación (PTA) y constantes individuales de velocidad para una serie de mecanismos de reacción, además de para llevar a cabo la discriminación en algunos modelos cinéticos estudiados. Los resultados obtenidos se exponen en tres bloques diferenciados. En un primer bloque se exponen los resultados obtenidos en la optimización de Parámetros Termodinámicos de Activación (PTA) en los siguientes mecanismos de reacción:



Los Parámetros Termodinámicos de Activación optimizados son los siguientes: E_{12} , A_{12} , ΔS_{12}^\ddagger , ΔH_{12}^\ddagger , E_{23} , A_{23} , ΔS_{23}^\ddagger y ΔH_{23}^\ddagger .

En un segundo bloque consideramos la determinación de las constantes de velocidad en el modelo de dos reacciones consecutivas mediante la metodología de Redes Neuronales Artificiales (en este caso no utilizaremos el AH ya que los resultados obtenidos únicamente con ANN son muy satisfactorios y al aplicar posteriormente el AGDC los resultados no se modifican).

En el tercer bloque se exponen los resultados obtenidos cuando se utilizan las metodologías ANN y el algoritmo de optimización AGDC para la discriminación de mecanismos de reacción, mediante la determinación de las constantes de velocidad de diferentes mecanismos que pudieran ser responsables del transcurso de una determinada reacción.

VII.2.- Determinación de Parámetros Termodinámicos de Activación en diferentes mecanismos de reacción mediante AH.

A lo largo de este apartado se exponen los resultados obtenidos en la determinación de los Parámetros Termodinámicos de Activación mediante el AH en diferentes mecanismos de reacción. La estructura de este apartado es la siguiente:

- En primer lugar se dedica un sub-apartado al análisis de las curvas T/t y α/t .
- A continuación se exponen los resultados obtenidos para el sistema $B_1 \xrightarrow{k_{12}} B_2$, considerando en primer lugar la determinación de los Parámetros de Arrhenius y a continuación los Parámetros de Eyring. En ambos casos se realiza el tratamiento de datos cinéticos no isotérmicos sintéticos y de datos experimentales.
- La exposición de los resultados obtenidos para el sistema $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ se lleva a cabo siguiendo una estructura análoga.

VII.2.1.- Análisis de las curvas T/t y de las curvas α/t

La aplicación del AH (ANN-AGDC) para la determinación de los Parámetros Termodinámicos de Activación (Parámetros de Arrhenius, E y A , y Parámetros de Eyring, ΔH^\ddagger y ΔS^\ddagger) se ha llevado a cabo a partir del tratamiento de datos procedentes de cinéticas no isotérmicas. La adquisición de datos cinéticos mediante un método no isotérmico supone la imposición de una variación continua de la temperatura en el transcurso de la cinética y puesto que la función temperatura vs tiempo (T vs t) debe satisfacer unas determinadas condiciones, es necesario realizar previamente un riguroso análisis de las curvas T/t. Asimismo en el caso de cinéticas no isotérmicas resulta conveniente utilizar como variable de trabajo la fracción remanente del reactivo B_j , α_j , por lo que es necesario analizar las curvas α/t .

VII.2.1.1.- Análisis de las curvas T/t

En condiciones no isotérmicas, la temperatura (T) varía con el tiempo (t) mediante la función T/t que tiene que satisfacer los siguientes requerimientos o condiciones:

- La función debe ser monótona creciente ya que, así se minimiza la gran diferencia existente en la velocidad de reacción entre el principio y el final de la misma al considerar condiciones isotérmicas. Así, en condiciones no isotérmicas con una función T/t creciente, la disminución de la velocidad de reacción debida al agotamiento de los reactivos, se va compensando en el transcurso de la misma con el aumento de la velocidad debido al incremento de la temperatura.

- La función tiene que ser accesible a la experimentación y reproducible en el laboratorio.

Teniendo en cuenta las condiciones que debe cumplir la función T/t , los tipos de funciones que hemos elegido entre las múltiples posibilidades que existen son las siguientes:

a) La función inversa de la temperatura que corresponde a la rama hiperbólica [III.4.27] que nos permite alcanzar, solo en el caso de la ecuación de Arrhenius, una solución matemática exacta para la ecuación diferencial de velocidad.

b) La función polinómica de grado n [III.4.33], que conduce a una ecuación diferencial de velocidad sin solución matemática exacta [III.4.34].

Además de los requisitos que debe cumplir la función T/t , es necesario tener en cuenta que la velocidad de calentamiento debe ser la adecuada para el intervalo de tiempo estudiado, en el cuál, pueden ser adquiridos los datos de las cinéticas no isotérmicas. Es necesario que ese perfil sea reproducible en el laboratorio. Analizando la rama hiperbólica, el intervalo de temperaturas que se considera adecuado es el comprendido entre 25°C (298.15K) y 45°C (318.15K). El intervalo de tiempo de calentamiento que hemos utilizado varía en función del modelo estudiado, intentando conseguir en ese intervalo que la concentración de la especie en juego se reduzca a una fracción igual a dos veces el tiempo de vida media. Hemos ensayado otros intervalos de temperaturas, con el mismo tiempo de calentamiento, obteniendo los distintos valores de “ m ”, de la rama hiperbólica (Figura VII.1.1). A continuación, se lleva a cabo la representación de los valores sintéticos T/t para la rama hiperbólica con los diferentes valores del parámetro “ m ”.

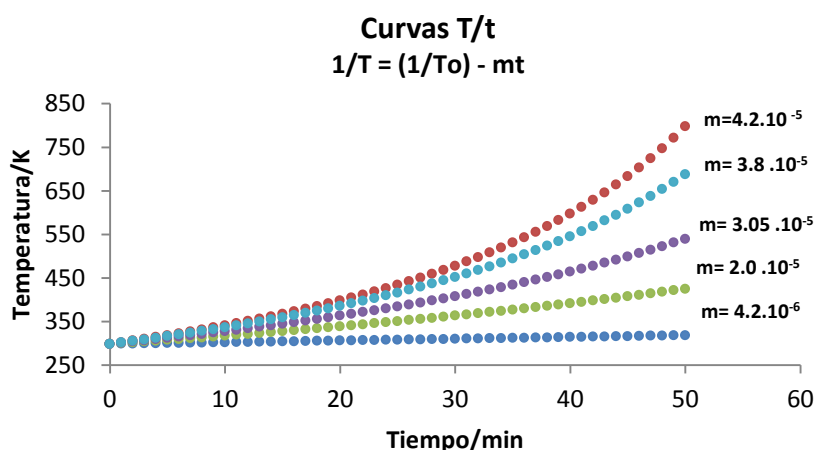


Figura VII.1.1: Representación gráfica de los valores de “ m ” de la rama hiperbólica con el mismo tiempo de calentamiento.

La representación gráfica de la Figura VII.1.1 muestra que, a medida que disminuye el valor de “ m ”, la concavidad de la rama hiperbólica va desapareciendo tendiendo a la linealidad. Por dicho motivo, el valor idóneo de “ m ” y que mejor satisface los requisitos de la función T/t es $4.20 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$. Los valores de los puntos T/t se ajustan a una función polinómica de

primer grado, de segundo grado y así progresivamente hasta polinomios de quinto grado. Tal como cabía esperar, se observa que los coeficientes de la variable independiente (t) poseen valores despreciables respecto al valor de a_1 a partir del término cuadrático. La Tabla VII.1.1 muestra los coeficientes a_0, a_1, \dots, a_5 obtenidos en el ajuste para los polinomios de grado 1, 3 y 5:

Grado del Polinomio	a_0	a_1	a_2	a_3	a_4	a_5
1	298.15	0.3734	-	-	-	-
2	298.15	0.3734	$5.0 \cdot 10^{-4}$	$6.0 \cdot 10^{-7}$	-	-
3	298.15	0.3734	$5.0 \cdot 10^{-4}$	$6.0 \cdot 10^{-7}$	$7.0 \cdot 10^{-10}$	$1.0 \cdot 10^{-12}$

Tabla VII.1.1: Valores de los coeficientes de las funciones polinómicas de primer, tercer y quinto grado respectivamente.

A la vista de los resultados, se puede concluir que para la rama hiperbólica T/t los valores experimentales se pueden ajustar a una recta: $T = 298.15 + 0.3734t$. Los valores que se obtienen ajustando directamente a una recta coinciden con los valores que provienen de la hipérbola. Como la hipérbola es muy tendida, se recorren los mismos puntos de la hipérbola con la recta. Por tanto los pares de valores T/t generados con la función inversa hiperbólica con un valor de $m = 4.20 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$ pueden ajustarse satisfactoriamente a una función lineal con valor de pendiente $\phi = 0.3734 \text{ Kmin}^{-1}$. Esto nos permite imponer en el laboratorio una velocidad de calentamiento lineal, con una pendiente de 0.3734 Kmin^{-1} , comenzando con $T = 298.15 \text{ K}$ ($t=0$). Así, se tiene la seguridad de que los puntos experimentales T/t coinciden con los puntos teóricos pertenecientes a la función hiperbólica que conduce a una solución exacta de la ecuación diferencial de velocidad.

VII.2.1.2.- Análisis de las curvas α/t

La aplicabilidad de la metodología propuesta (ANN-AGDC) se comprueba mediante el tratamiento de datos sintéticos generados para valores conocidos de los Parámetros Termodinámicos de Activación (PTA), correspondientes a un determinado mecanismo de reacción. Es necesario seleccionar unos valores generales para los PTA, recurriendo a referencias bibliográficas del mecanismo considerado y con los mismos, generar las curvas cinéticas no isotérmicas α/t .

Consideremos el mecanismo más sencillo: $B_1 \xrightarrow{k_{12}} B_2$, en el caso de los parámetros termodinámicos de la ecuación de Arrhenius, A_{12} y E_{12} , se toman como valores generales $A_{12} = 10^6 \text{ min}^{-1}$ y $E_{12} = 4.1840 \times 10^4 \text{ Jmol}^{-1}$ (76). La fracción remanente de la concentración de reactivo se determina mediante la expresión [III.4.25]:

$$\alpha_1 = \frac{[B_1]}{[B_1]_0}$$

Es conveniente que el valor de α (en este caso concreto α_1) fluctúe en el intervalo de 0.25 a 1. Se utiliza como criterio razonable estudiar un intervalo de reacción en el que el tiempo transcurrido sea dos veces el tiempo de vida media, ya que, en ese momento, habrá transcurrido el 75% de la reacción. En este tipo de cinéticas no isotérmicas la velocidad de la reacción se ve afectada por dos efectos:

- La **concentración** de los reactivos: Al transcurrir la reacción, va disminuyendo la concentración del reactivo y por consiguiente, se produce un descenso en la velocidad de la reacción con el tiempo. El progreso de la cinética se va ralentizando.

- La **temperatura**: Un aumento de la temperatura favorece un aumento en la velocidad de reacción, por lo tanto, el efecto que provoca la natural disminución de la concentración de reactivo se contrarresta ya que son efectos contrarios. Éste es el motivo por el que siempre han de elegirse funciones monótonamente crecientes. Si se produjera un descenso de la temperatura, a la vez que un descenso de la concentración de los reactivos, la velocidad de reacción sería demasiado lenta, con los problemas derivados de este hecho.

Los perfiles de las curvas cinéticas no isotérmicas (α/t) presentan varios tramos cóncavos mientras que las curvas isotérmicas son siempre convexas. Este hecho se explica porque en las cinéticas no isotérmicas existen los dos efectos opuestos ya mencionados que afectan a la velocidad de la reacción: el incremento de la temperatura con el tiempo aumenta la velocidad de reacción, y la disminución de la concentración de reactivo que se va agotando, la disminuye.

De acuerdo con lo expuesto con anterioridad, se observan los siguientes perfiles para las curvas cinéticas α/t :

- Si la influencia de la concentración de reactivo y la temperatura fuera la misma daría lugar a una recta.
- Si predomina el efecto de la concentración sobre el efecto de la temperatura se producirá una curva con perfil convexo.
- Si por el contrario el efecto de la temperatura predomina sobre el efecto de la concentración, se producirá una curva con perfil cóncavo.

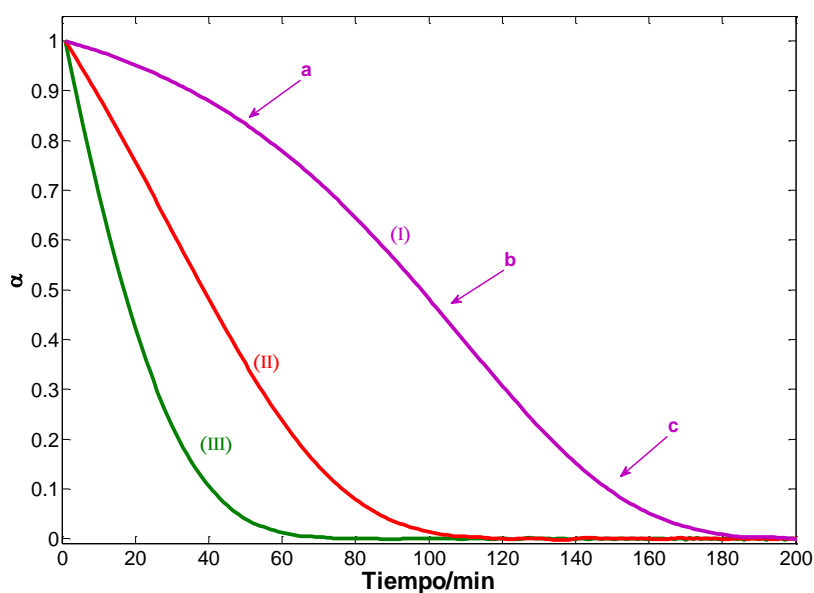


Figura VII.2.1: Representación gráfica de varios perfiles de las curvas no isotérmicas α/t .

En la gráfica se observa:

- La curva I es una sigmoidea correspondiente a una cinética no isotérmica en la que se observan tres tramos: en el tramo a), la curva es convexa, ya que predomina el efecto de la temperatura; en el tramo c), la curva es cóncava, puesto que predomina el efecto de la disminución de la concentración de reactivo; y el tramo intermedio b), la curva es prácticamente lineal pues ambos efectos se equilibran.
- La curva II presenta un perfil lineal pues los dos efectos están equilibrados durante toda la cinética.
- La curva III es cóncava en su totalidad, ya que, corresponde a una cinética isotérmica clásica, donde solo existe el efecto de la concentración.

VII.2.2.- Mecanismo I. $B_1 \xrightarrow{k_{12}} B_2$

Consideramos en primer lugar la aplicación del AH para la determinación de los PTA del mecanismo de reacción I (E_{12} y A_{12} y ΔS_{12}^\ddagger y ΔH_{12}^\ddagger). En ambos casos se realiza el tratamiento de datos cinéticos no isotérmicos sintéticos y experimentales y se analiza en primer lugar el proceso de *Predicción* mediante ANN y a continuación el proceso de *Optimización Matemática* mediante el algoritmo AGDC.

VII.2.2.1.- Determinación de los parámetros E_{12} y A_{12}

En este apartado se exponen los resultados obtenidos en la determinación de los parámetros de la ecuación de Arrhenius (E_{12} y A_{12}) mediante la aplicación del Algoritmo Híbrido (ANN-AGDC). En primer lugar se realiza el tratamiento de datos sintéticos con objeto de analizar la aplicabilidad de la metodología y una vez comprobada la aplicabilidad se lleva a cabo el tratamiento de datos experimentales. Puesto que el AH se basa en la aplicación de dos metodologías diferentes, ANN y AGDC, se ofrecen los resultados obtenidos individualmente con cada una de ellas en ambos casos.

VII.2.2.1.1- Tratamiento de datos sintéticos

VII.2.2.1.1.a- Predicción de E_{12} y A_{12} mediante ANN

i) Diseño de Experimentos

Para calcular la Energía de Activación (E_{12}) y el Factor Preexponencial (A_{12}) de una o varias cinéticas mediante ANN, es necesario realizar el correspondiente Diseño de Experimentos teniendo una estimación previa o algún indicio de los valores de E_{12} y A_{12} que nos proporcionen, al menos, sus órdenes de magnitud. Para ello, tenemos que acudir a algún método gráfico que permita determinar los órdenes de magnitud absoluto y relativo de las mismas o bien, utilizar alguna propiedad o característica cinética del modelo de reacción. Los valores para E_{12} y A_{12} con los que se generarán los datos sintéticos para el mecanismo que estamos considerando, han sido tomados de bibliografía ⁽⁷⁶⁾ y son los siguientes: $E_{12} = 4.1840 \cdot 10^4 \text{ Jmol}^{-1}$ y $A_{12} = 1.10^6 \text{ min}^{-1}$.

Para llevar a cabo el Diseño de Experimentos se consideran modelos estadísticos clásicos cuyo objetivo es averiguar si determinados *factores* influyen de manera cualitativa en alguna variable del sistema o proceso de estudio. Si se detecta la influencia de algún *factor*, hay que cuantificar dicha influencia. La realización de un Diseño de Experimentos (ED) es crucial para asegurar el éxito del proceso *Entrenamiento* de las Redes Neuronales. En nuestro caso, en el ED están implicados dos *factores*, correspondientes a E_{12} y A_{12} de la ecuación de Arrhenius, cuyas *respuestas* son los datos no isotérmicos cinéticos de la base de las curvas de entrada (α_1/t). Teniendo en cuenta las características del cómputo de ANN del sistema cinético no isotérmico en estudio, hemos escogido como el óptimo, el ED correspondiente al denominado *Central Star Composite Experimental Design* (CSCED) cuya distribución se encuentra en la Figura VII.2.2.

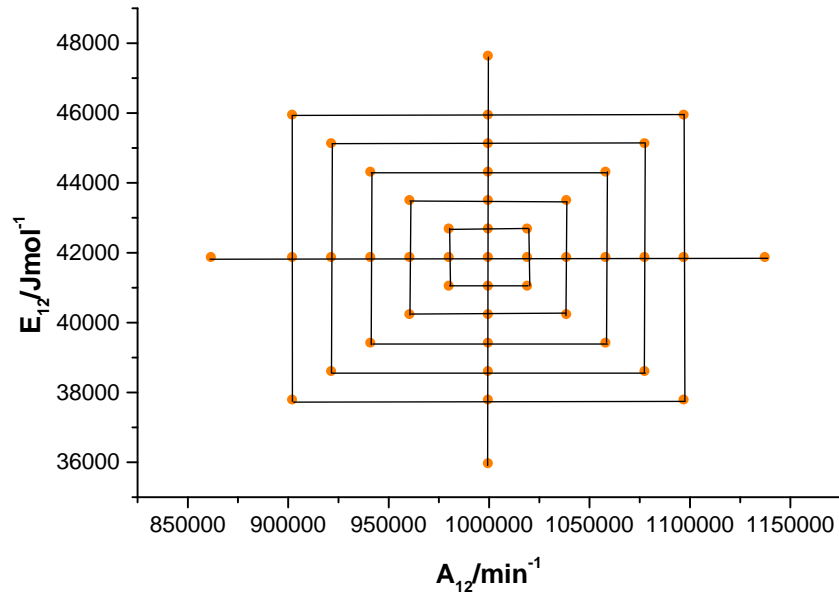


Figura VII.2.2: Diseño de Experimentos (CSCED) constituido por 2 factores (A_{12} y E_{12}) y 45 puntos distribuidos en 9 niveles (puntos del 1-37) y 4 subniveles (puntos del 38 - 45) que participan en la base de curvas de la matriz de *Inputs* utilizada para el proceso de *Entrenamiento* de la Red Neuronal para el modelo cinético $B_1 \xrightarrow{k_{12}} B_2$.

ii) *Entrenamiento*

Una vez que hemos generado las 45 curvas sintéticas correspondientes a la cinética no isotérmica (*Inputs*), considerando una variación de la temperatura hiperbólica, llevamos a cabo el proceso de *Entrenamiento* de la Red Neuronal Artificial (ANN) con el fin de encontrar la Arquitectura más adecuada que nos permita determinar la Energía de activación, E_{12} y el Factor preexponencial A_{12} . Para determinar esta Arquitectura óptima de la Red Neuronal se realizó el *Entrenamiento* variando el número de nodos y la Configuración de las capas Ocultas (*Hidden*) sistemáticamente, considerando todas las opciones posibles (Tabla VII.2.2).

ANN <i>Artificial Neural Networks</i>					
Config.	$A_{12}/10^6 \text{ min}^{-1}$	$E_{12}/10^4 \text{ Jmol}^{-1}$	MSE	Des. % (A_{12})	Des. % (E_{12})
14/10	0.98000	4.1933	$2.8601 \cdot 10^{-6}$	1.9990	$-2.2333 \cdot 10^{-1}$
15/10	0.98018	4.1871	$2.7120 \cdot 10^{-6}$	1.9908	$-7.4611 \cdot 10^{-2}$
7/10/10	0.98005	4.1871	$3.1803 \cdot 10^{-6}$	1.9965	$-7.3811 \cdot 10^{-2}$
9/10/10	0.98015	4.1950	$1.9500 \cdot 10^{-6}$	1.9995	$-2.6310 \cdot 10^{-1}$
10/9/10	0.98005	4.1961	$3.3010 \cdot 10^{-6}$	1.9995	$-2.8851 \cdot 10^{-1}$
10/10/7	0.98048	4.1917	$2.7504 \cdot 10^{-6}$	1.9628	$-1.8477 \cdot 10^{-1}$
10/10/13	0.97951	4.1938	$2.7505 \cdot 10^{-6}$	2.0491	$-2.3318 \cdot 10^{-1}$
10/12/10	0.97992	4.1885	$2.6502 \cdot 10^{-6}$	2.0002	$-1.0825 \cdot 10^{-1}$
10/20/10	0.98068	4.1893	$2.8710 \cdot 10^{-7}$	1.9438	$-1.2733 \cdot 10^{-1}$
11/10/10	0.98005	4.1923	$2.5399 \cdot 10^{-6}$	1.9955	$-1.9933 \cdot 10^{-1}$
20/10/10	0.97954	4.1935	$2.7198 \cdot 10^{-6}$	2.0514	$-2.2675 \cdot 10^{-1}$

Tabla VII.2.1: Valores de los PTA y desviaciones obtenidas después de la aplicación de ANN para el proceso de *Entrenamiento* de un gran conjunto de redes neuronales considerando todas las posibles configuraciones para Arquitecturas con 2 y 3 *capas Ocultas*.

Así, tras la experimentación comprobamos que la Red Neuronal más adecuada consta de 3 capas (Estructura) y dentro de esa capa el número de nodos óptimos es 10/20/10 (Configuración), utilizando en todos los casos el mismo porcentaje de curvas para el *Entrenamiento*, *Validación* y *Test*, (80/10/10). La manera de confirmar que esa es la Configuración óptima consiste en determinar el valor de MSE y la que posea el menor es la que resulta adecuada para realizar la *Predicción* de la Energía de activación y el Factor preexponencial. Otra manera de visualizar si el resultado obtenido es el adecuado consiste en las Regresiones lineales *Outputs/Targets* de la Configuración que consideramos óptima, y los perfiles de las curvas de *Entrenamiento*, *Validación* y *Test* de la misma. Para concluir que los resultados son satisfactorios, evaluamos el cumplimiento de las siguientes condiciones:

- Ecuaciones lineales (pendiente ≈ 1 , ordenada ≈ 0).
- Coeficiente de regresión ($R \approx 1$).
- Representaciones gráficas adecuadas (dispersión de las curvas y los puntos)...etc.

En la evaluación de los resultados obtenidos en el proceso de *Entrenamiento* de la ANN, es importante analizar los resultados de la regresión lineal de las matrices de *Outputs vs Targets*. En la Figura VII.2.3 se muestra la representación gráfica de estos resultados, realizada de forma conjunta para ambos parámetros (A_{12} y E_{12}). Como poseen órdenes de magnitud muy diferentes, $E_{12} = 10^4$ y $A_{12} = 10^6$, sus valores se encuentran obviamente muy alejados dentro de la gráfica conjunta. En el caso de las desviaciones del parámetro A_{12} , aparenta existir una buena linealidad. En el caso de las desviaciones del parámetro E_{12} , todos los valores de dispersión se superponen y producen un único punto, esto es debido a las diferentes magnitudes de las escalas de ambos parámetros. Esto enmascara los verdaderos efectos de la dispersión haciendo que los parámetros se ajusten a una única recta de regresión pudiendo ser considerada aceptable esta gráfica en principio.

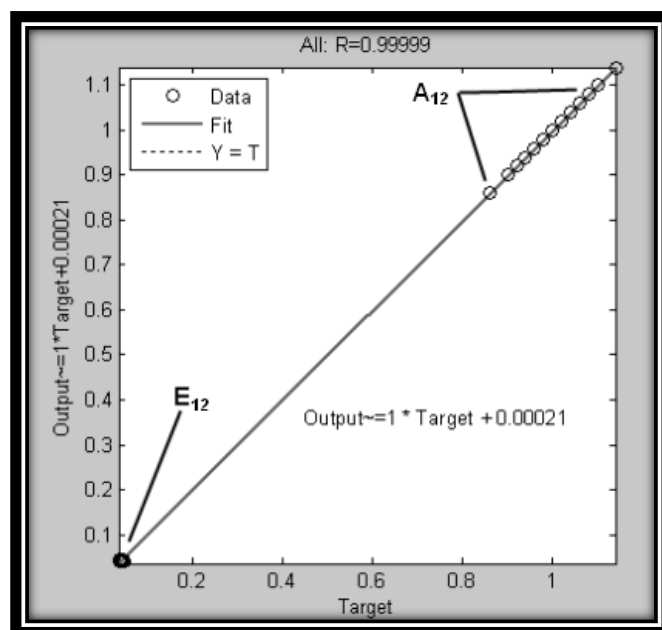


Figura VII.2.3: Regresión lineal de las matrices de *Outputs vs Targets*, cuando ambos parámetros (A_{12} y E_{12}) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* (50, 3, 2).

Sin embargo, cuando se consideran las regresiones *Outputs/Targets* de cada uno de los parámetros por separado, en dos gráficas distintas, utilizando las escalas apropiadas, para el caso de A_{12} se corrobora la linealidad observada en la Figura VII.2.3, sin embargo, en el caso de E_{12} se detecta una enorme dispersión y una linealidad inaceptable tal y como puede verse en la Figura VII.2.4.

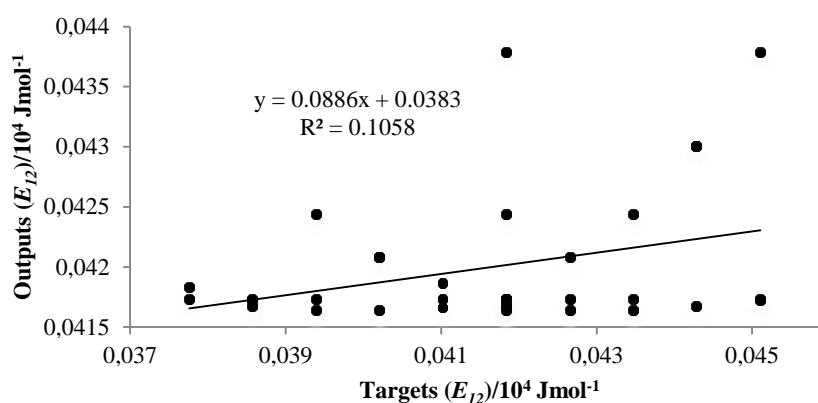


Figura VII.2.4: Regresión lineal de las matrices de *Outputs vs Targets*, exclusivamente en el caso de E_{12} para el proceso de *Entrenamiento* de la ANN con una arquitectura de 3 *hidden layers* (50, 3, 2).

La superposición de los puntos observada en la Figura VII.2.3 se amplía constatando que no hay ninguna linealidad tal como en un principio aparentemente se creía. Se obtienen valores de R^2 (0.1058), pendiente (0.0886) y ordenada en el origen (0.0383) que ponen de manifiesto una correlación muy pobre entre los valores de *Outputs* y *Targets* que hacen

inaceptables los valores de los parámetros determinados por ANN como valores finales. Por tanto, se hace necesario un tratamiento posterior que se lleva a cabo con la aplicación de la segunda metodología de la que consta el Algoritmo Híbrido (AH), es decir el algoritmo de Optimización Matemática AGDC, obteniendo los valores definitivos de los parámetros a determinar.

iii) *Predicción*

Una vez que se ha llevado a cabo el *Entrenamiento* de la Red Neuronal Artificial y se dispone de la Arquitectura óptima, se puede utilizar la Red para realizar la *Predicción* de los valores de la E_{12} y A_{12} , a partir de datos cinéticos de fracción remanente α_1 , en el caso de que su valor sea desconocido en determinadas experiencias cinéticas. Tal y como hemos señalado anteriormente, hemos realizado el *Entrenamiento* de un gran colectivo de Redes Neuronales considerando un gran número de Configuraciones posibles para Arquitecturas con 2 y 3 capas Ocultas (*Hidden*) descartando las de 1 única capa. Dado el elevado número de casos ensayados, hemos representado únicamente los mejores resultados de las configuraciones más significativas de la curva correspondientes al punto central de las 45 que componen el ED por ser el más importante. Es necesario cuantificar la Desviación Estándar individual de cada parámetro obtenido tras la aplicación de ANN con respecto al valor real (valores de *Outputs* y *Targets* del conjunto de las 45 curvas). En las columnas 2, 3, 4 y 5 de la Tabla VII.2.2 se muestran los valores de *Outputs* de E_{12} y A_{12} y de las desviaciones en % ("*Des. %*") respecto a los que sirvieron para generar los datos. Los valores obtenidos para los errores de los parámetros son inaceptables por lo es necesario mejorar los resultados mediante la optimización mediante la aplicación de la segunda etapa del Algoritmo Híbrido (AH) que constituye la aplicación de AGDC.

VII.2.2.1.1.b.- Optimización de E_{12} y A_{12} mediante AGDC

Los valores de los PTA que han sido obtenidos mediante el proceso de *Predicción* realizado con ANN, son sometidos a un proceso de mejora y refinamiento mediante la aplicación del programa **KINNOISOT(AGDC)**. Este programa está basado en el algoritmo de Optimización Matemática AGDC, cuyas características computacionales se han expuesto en el capítulo IV, y que utiliza como estimaciones iniciales los valores obtenidos por ANN.

La Tabla VII.2.2 muestra los resultados obtenidos para los valores de E_{12} y A_{12} obtenidos con ANN y los obtenidos tras la aplicación del Algoritmo Híbrido. En la primera parte de la Tabla se muestran los valores de *Outputs* de E_{12} y A_{12} obtenidos con ANN y las desviaciones en % ("*Des. %*") de estos valores con respecto a los que sirvieron para generar los datos. En la 2ª parte aparecen los valores de E_{12} y A_{12} optimizados con AGDC utilizando como estimaciones iniciales los valores procedentes del proceso de *Predicción* con ANN, además se

muestran las desviaciones en % de los valores reales (“Des. %”). Se observa claramente la mejora que consigue la aplicación completa del AH al comparar los valores de las desviaciones (columnas 8ª y 9ª) de ambos parámetros, con los valores correspondientes tras la aplicación de ANN (columnas 4ª y 5ª).

Config.	ANN				(ANN - AGDC) AH			
	$A_{I2}/10^6$ min ⁻¹	$E_{I2}/10^4$ Jmol ⁻¹	Des. % (A_{I2})	Des. % (E_{I2})	$A_{I2}/10^6$ min ⁻¹	$E_{I2}/10^4$ Jmol ⁻¹	Des. % (A_{I2})	Des. % (E_{I2})
14/10	0.98000	4.1933	1.9990	-2.2333.10 ⁻¹	1.0070	4.1859	-7.0379.10 ⁻¹	-4.5312.10 ⁻²
15/10	0.98018	4.1871	1.9908	-7.4611.10 ⁻²	0.99555	4.1829	4.4511.10 ⁻¹	2.5321.10 ⁻²
7/10/10	0.98005	4.1871	1.9965	-7.3811.10 ⁻²	0.99555	4.1829	4.5479.10 ⁻¹	2.5925.10 ⁻²
9/10/10	0.98015	4.1950	1.9995	-2.6310.10 ⁻¹	1.0101	4.1867	-1.0104	-6.4011.10 ⁻²
10/9/10	0.98005	4.1961	1.9995	-2.8851.10 ⁻¹	1.0120	4.1872	-1.2051	-7.5811.10 ⁻²
10/10/7	0.98048	4.1917	1.9628	-1.8477.10 ⁻¹	1.0043	4.1852	-4.2601.10 ⁻¹	-2.8312.10 ⁻²
10/10/13	0.97951	4.1938	2.0491	-2.3318.10 ⁻¹	1.0075	4.1860	-7.5211.10 ⁻¹	-4.8312.10 ⁻²
10/12/10	0.97992	4.1885	2.0002	-1.0825.10 ⁻¹	0.99811	4.1836	1.8801.10 ⁻¹	9.4156.10 ⁻³
10/20/10	0.98068	4.1893	1.9438	-1.2733.10⁻¹	0.99999	4.1841	9.4010.10⁻³	-1.5020.10⁻³
11/10/10	0.98005	4.1923	1.9955	-1.9933.10 ⁻¹	1.0052	4.1854	-5.2122.10 ⁻¹	3.4111.10 ⁻²
20/10/10	0.97954	4.1935	2.0514	-2.2675.10 ⁻¹	1.0500	4.1866	-7.0168.10 ⁻¹	-4.5221.10 ⁻²

Tabla VII.2.2: Valores de los PTA y desviaciones obtenidas después de la aplicación de ANN y el algoritmo completo AH para el proceso de *Entrenamiento* de un gran conjunto de Redes Neuronales considerando todas las posibles Configuraciones para Arquitecturas con 2 y 3 *capas Ocultas*.

Cuando se realiza una representación conjunta (Figura VII.2.5) de los Residuales de α_1 calculados con los valores de los parámetros obtenidos mediante la aplicación de ANN y los conseguidos tras la aplicación completa del Algoritmo Híbrido (AH), se aprecia con claridad la mejora de los resultados. Para el cálculo de los residuales se han utilizado las siguientes expresiones:

$$[Res(\alpha)_{ANN} = \alpha_{sint} - \alpha_{cal}(ANN)] \quad [VII.2.1]$$

$$[Res(\alpha)_{HA} = \alpha_{sint} - \alpha_{cal}(AH)] \quad [VII.2.2]$$

donde α_{sint} corresponden a los valores de α sintéticamente determinados.

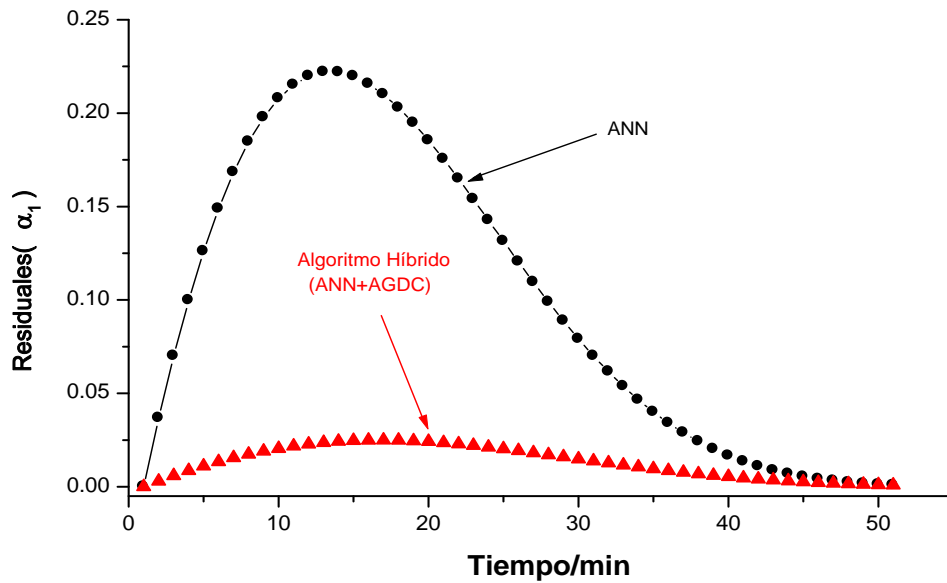


Figura VII.2.5: Figura de las residuales de α calculados a partir de la ecuación [VII.2.1] con los valores de los parámetros obtenidos mediante la aplicación de la ANN y los obtenidos a partir de la ecuación [VII.2.2] después de la aplicación del algoritmo completo AH.

Teniendo en cuenta los valores obtenidos para las desviaciones, podemos constatar que la Arquitectura óptima de la Red Neuronal es (50, 3, 2) constituida por 3 capas Ocultas (“Hidden”) con una configuración 10/20/10. Esta Arquitectura ha sido la que hemos utilizado para realizar los procesos posteriores de *Predicción* con ANN y Optimización Matemática mediante AGDC.

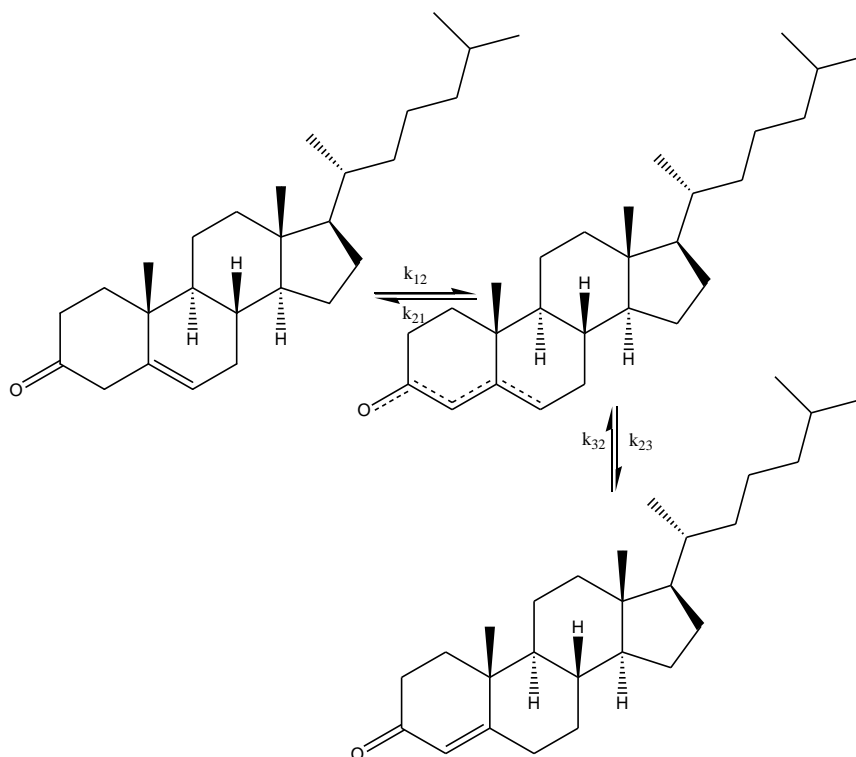
Por último hemos aplicado el Algoritmo Híbrido (AH) para determinar los valores de los parámetros A_{12} y E_{12} correspondientes a experiencias cinéticas no isotérmicas comprendidas dentro de los intervalos del ED, pero distintas a las utilizadas en el *Entrenamiento*. El proceso de *Predicción* se llevó a cabo con la Red Neuronal óptima para cuatro curvas cinéticas no isotérmicas, con lo que se lleva a cabo la *Predicción* de 4 pares de valores de parámetros A_{12}/E_{12} . Los resultados obtenidos para la *Predicción* de esos 4 pares de valores A_{12}/E_{12} mediante ANN, son sometidos a un proceso posterior de optimización mediante la aplicación del algoritmo AGDC, obteniendo así los valores óptimos de dichos parámetros. Teniendo en cuenta que los porcentajes de desviaciones (“% Des.”) poseen una magnitud similar al de las desviaciones del proceso de *Entrenamiento* de la Red Neuronal óptima, podemos aceptar los valores optimizados de A_{12} y E_{12} como definitivos para los 4 procesos de *Predicción*. Los resultados obtenidos se muestran en la Tabla VII.2.3.

	(ANN	-	AGDC)	AH	
Valores Reales (A_{12})/ 10^6 min^{-1}	Valores Reales (E_{12})/ 10^4 Jmol^{-1}	Valores Optimizados (A_{12})/ 10^6 min^{-1}	Valores Optimizados (E_{12})/ 10^4 Jmol^{-1}	% Des. (A_{12})	% Des. (E_{12})
0.98000	4.1891	0.98970	4.1865	$-9.9077 \cdot 10^{-1}$	$-6.1591 \cdot 10^{-2}$
0.96000	4.1944	0.97920	4.1891	-2.0002	$-1.2414 \cdot 10^{-1}$
0.98000	4.1891	0.98000	4.1891	$-2.0089 \cdot 10^{-4}$	-2.1111
0.85910	4.1808	0.85371	4.1824	$6.1177 \cdot 10^{-1}$	$4.8408 \cdot 10^{-2}$

Tabla VII.2.3: Valores de los PTA y desviaciones obtenidas después de la aplicación del algoritmo completo AH para el proceso de Predicción cuando ha sido aplicada la Red con una Arquitectura óptima (50, 3, 2) y Configuración (20/10/10).

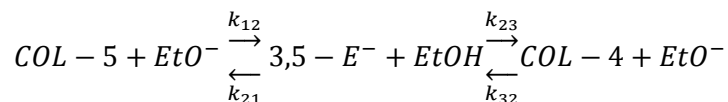
VII.2.2.1.2.- Tratamiento de datos experimentales

Una vez que hemos comprobado la aplicabilidad del AH para la determinación de E_{12} y A_{12} en el mecanismo de reacción $B_1 \xrightarrow{k_{12}} B_2$ con datos cinéticos generados sintéticamente, hemos aplicado el AH para el tratamiento de datos experimentales. Corresponden a la reacción de isomerización de colest-5-en-3-ona a colest-4-en-3-ona catalizada por etóxido sódico. El esteroide colest-5-en-3-ona (COL-5) se transforma en su isómero colest-4-en-3-ona (COL-4) (Esquema VII.2.1) en medio etanol absoluto, a través de un proceso reversible que puede ser catalizado por ácidos, bases y determinadas enzimas específicas ^(70-71,157).



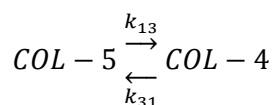
Esquema VII.2.1: Reacción de isomerización de colest-5-en-3-ona a colest-4-en-3-ona catalizada por etóxido sódico.

De acuerdo con las evidencias experimentales ^(70-71,157) se puede proponer un mecanismo de reacción para la isomerización de colest-5-en-3-ona (COL-5) a colest-4-en-3-ona (COL-4) catalizada por EtO^- en medio etanol, constituido por dos pasos reversibles en los cuales el 3,5-dienolato ($3,5 - E^-$) se considera una especie intermedia altamente reactiva.



Esquema VII.2.2: Reacción simplificada en la que tiene lugar el proceso de isomerización de $COL - 5$ a $COL - 4$.

La reacción global de isomerización catalizada por los iones etóxido, considerando exclusivamente reactivos, productos y catalizador, se puede escribir como:



Esquema VII.2.3: Reacción global de isomerización de colest-5-en-3-ona a colest-4-en-3-ona.

Aunque la reacción es reversible, aplicamos el AH para el estudio de la reacción en el sentido directo, es decir, en sentido de conjugación, cuando se parte de colest-5-en-3-ona y etóxido sódico. Hemos realizado un estudio exhaustivo de la cinética no isotérmica para lograr la determinación de los Parámetros Termodinámicos de Activación experimentales estudiando la influencia que la temperatura ejerce en la cinética global de la reacción. Los datos cinéticos obtenidos a diversas temperaturas (desde 288.15K hasta 308.15K) van a ser correlacionados adecuadamente mediante las ecuaciones correspondientes derivadas de las teorías de Arrhenius y Eyring.

VII.2.2.1.2.a- Predicción de E_{13} y A_{13} mediante ANN

i) Diseño de Experimentos

Para determinar E_{13} y A_{13} mediante ANN es necesario realizar el correspondiente Diseño de Experimentos a partir de una estimación previa o algún indicio de sus valores que nos proporcionen, sus órdenes de magnitud. Los valores experimentales para E_{13} y A_{13} , tomados de bibliografía ^(70-71,157), correspondientes a la reacción $COL - 5 \rightarrow COL - 4$, son los siguientes: $E_{13} = 6.28 \cdot 10^4 \text{ Jmol}^{-1}$ y $A_{13} = 1.96 \cdot 10^{10} \text{ min}^{-1}$.

En este caso, en el ED están implicados dos factores E_{13} y A_{13} correspondientes a la ecuación de Arrhenius, cuyas respuestas son los datos no isotérmicos cinéticos de la base de las curvas de Inputs (α_1/t). El ED que hemos seleccionado como óptimo es el *Central Star Composite Experimental Design* (CSCED) cuya distribución se encuentra en la Figura VII.2.6.

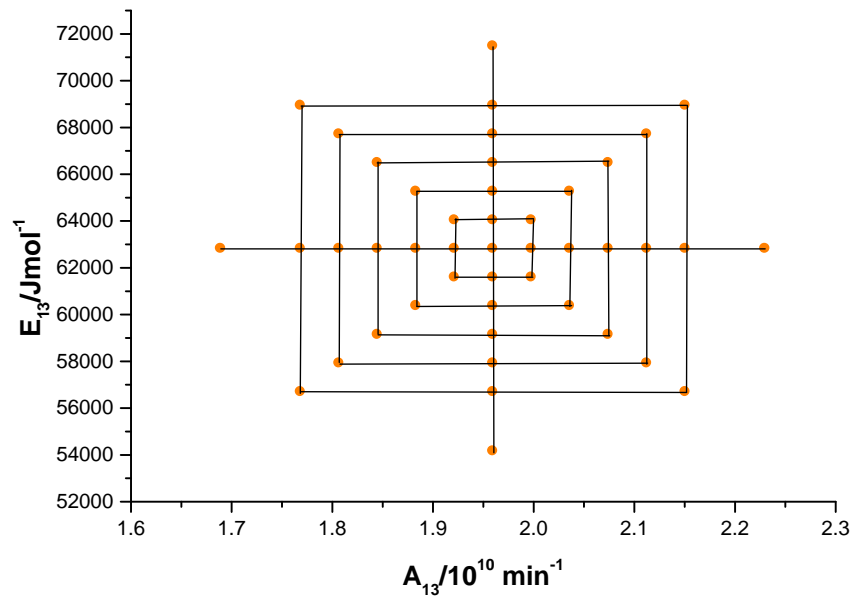


Figura VII.2.6: Diseño de Experimentos (CSCED) constituido por 2 factores (A_{13} y E_{23}) y 45 puntos distribuidos en 9 niveles (puntos del 1-37) y 4 subniveles (puntos del 38 - 45) que participan en la base de curvas de la matriz de *Inputs* utilizada para el proceso de *Entrenamiento* de la Red Neuronal.

ii) *Entrenamiento*

Una vez que hemos generado las 45 curvas experimentales no isotérmicas (*Inputs*) considerando una variación de la temperatura hiperbólica, llevamos a cabo el proceso de *Entrenamiento* de la Red Neuronal Artificial (ANN) con el fin de encontrar la más adecuada para realizar el proceso de *Predicción* de E_{13} y A_{13} . Para determinar la Arquitectura óptima de la Red Neuronal se realizó el *Entrenamiento* variando el número de nodos y la configuración de las capas Ocultas (*Hidden*) sistemáticamente, considerando una amplia casuística de las opciones posibles y utilizando en todos los casos la misma relación de porcentajes de curvas para el *Entrenamiento*, *Validación* y *Test* (80/10/10). Tras la experimentación comprobamos que la Red Neuronal óptima consta de 3 capas (Estructura) y con una Configuración 10/7/10.

Cuando se representan conjuntamente las regresiones *Outputs/Targets* de A_{13} y E_{13} (Figura VII.2.7) se puede comprobar que se encuentran muy separados dentro de la gráfica, como consecuencia de que presentan órdenes de magnitud muy diferentes ($E_{13} \approx 10^4$ y $A_{13} \approx 10^{10}$). En el caso de las desviaciones de A_{13} existe, aparentemente, una pésima linealidad y en el caso de las desviaciones de E_{13} , todos los valores se superponen y producen un único punto, esto es debido a las diferentes magnitudes de las escalas de ambos parámetros (Figura VII.2.7), lo cual enmascara los verdaderos efectos de la dispersión haciendo que los parámetros se ajusten a una buena recta de regresión siendo considerada esta gráfica aparentemente aceptable.

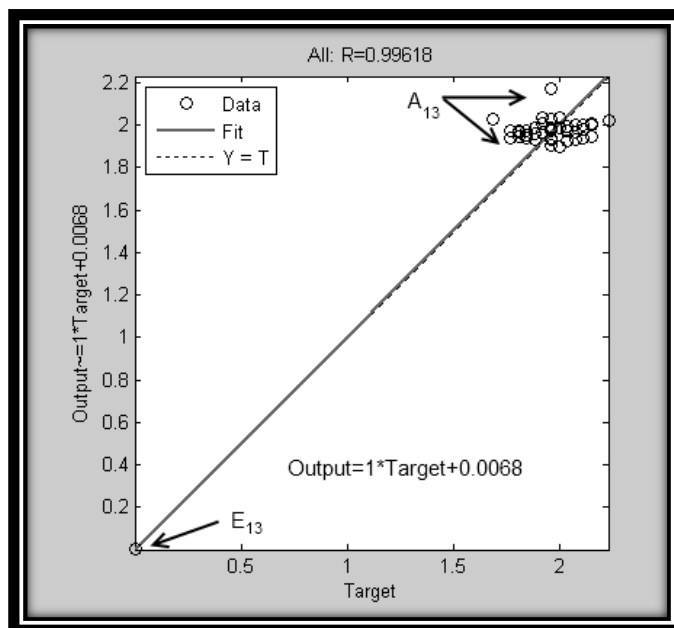


Figura VII.2.7: Regresión lineal de las matrices de *Outputs vs Targets*, cuando ambos parámetros (E_{13}/A_{13}) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* (25, 3, 2).

Si tenemos en cuenta las regresiones *Outputs/Targets* por separado se observa que existe una gran dispersión en ambos parámetros (Figuras VII.2.8.a y VII.2.8.b), esto hace que los resultados de la aplicación únicamente de ANN sean inaceptables, por tanto es necesario utilizar otro algoritmo, en nuestro caso AGDC, para obtener unos valores finales de A_{13} y E_{13} satisfactorios.

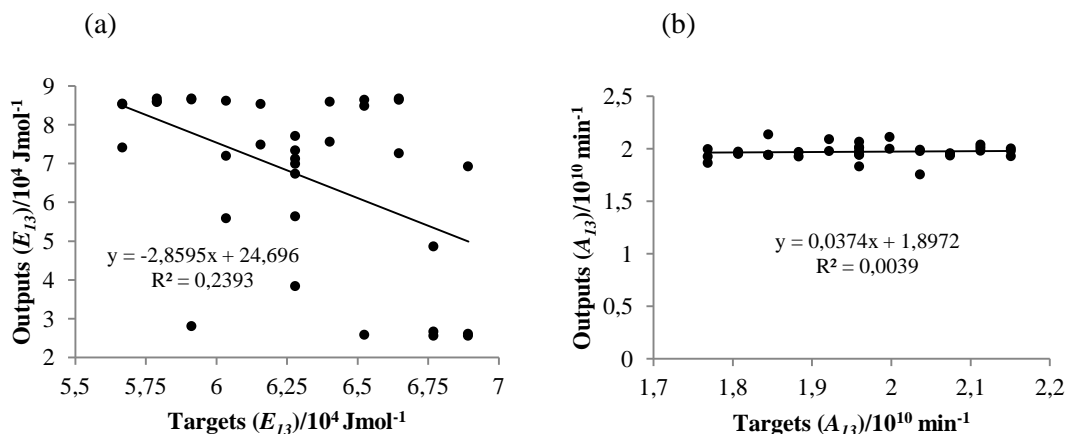


Figura VII.2.8.a y b: Regresión lineal de las matrices de *Outputs vs Targets*, exclusivamente en el caso de E_{13} (a) y A_{13} (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* (25, 3, 2).

iii) Predicción

La ANN óptima se utiliza para realizar el proceso de *Predicción* de E_{13} y A_{13} a partir de datos cinéticos experimentales, para determinadas experiencias cinéticas en las que su valor es desconocido. Hemos realizado el proceso de *Entrenamiento* de un gran conjunto de Redes Neuronales considerando todas las Configuraciones posibles para las Arquitecturas con 2 y 3

capas Ocultas. En la Tabla VII.2.4 se muestran los resultados más relevantes. Todos ellos fueron obtenidos para las Configuraciones más significativas correspondientes al punto central de las 45 curvas del ED. El par de valores de *Outputs*, E_{13}/A_{13} , y las correspondientes desviaciones (“Des. %”) se muestran en la primera parte de la Tabla VII.2.4. Los valores obtenidos para los errores de los parámetros indican que es necesario mejorar los resultados mediante la optimización posterior realizada en la segunda etapa del Algoritmo AGDC.

VII.2.2.1.2.b- Optimización de E_{13} y A_{13} mediante AGDC

Los valores de E_{13} y A_{13} obtenidos en el proceso de *Predicción* tras la aplicación de ANN, han sido utilizados como estimaciones iniciales en el proceso de optimización llevado a cabo mediante la aplicación del programa KINNOISOT(AGDC). La segunda parte de la Tabla VII.2.4 muestra los valores de los parámetros optimizados después de aplicar AGDC y las desviaciones de los valores reales (“Des. %”). Después de completar toda la aplicación del AH se observa una evidente mejora de resultados comparando los valores de las desviaciones (columnas octava y novena) de ambos parámetros con los calculados por la aplicación ANN (columnas cuarta y quinta). Teniendo en cuenta los valores obtenidos para las desviaciones, podemos corroborar que la Arquitectura óptima de la red neuronal es (25, 3, 2) constituida por 3 capas Ocultas con una Configuración 10/7/10. Esta Arquitectura ha sido la que hemos utilizado para realizar los procesos posteriores de *Predicción* con ANN y Optimización mediante AGDC.

Config.	ANN				(ANN - AGDC)			AH
	$A_{13}/10^{10}$ min ⁻¹	$E_{13}/10^4$ Jmol ⁻¹	Des. % (A_{13})	Des. % (E_{13})	$A_{13}/10^{10}$ min ⁻¹	$E_{13}/10^4$ Jmol ⁻¹	Des. % (A_{13})	Des. % (E_{13})
10/14	1.9299	6.0621	1.5351	3.4686	1.9326	6.2762	1.3986	$6.0026 \cdot 10^{-2}$
8/10/10	1.9201	6.4747	2.0318	-3.1006	1.9437	6.2750	$8.3007 \cdot 10^{-1}$	$7.9505 \cdot 10^{-2}$
10/7/10	1.9444	5.7361	$7.9334 \cdot 10^{-1}$	8.6607	1.9577	6.2781	$1.2007 \cdot 10^{-1}$	$3.0918 \cdot 10^{-2}$
10/10/11	1.9449	6.5542	$7.6889 \cdot 10^{-1}$	-4.3663	1.9463	6.5410	$6.9693 \cdot 10^{-1}$	-4.1568
10/10/12	2.1382	5.7873	-9.0962	7.8449	1.9673	6.3012	$3.7003 \cdot 10^{-1}$	$-3.3752 \cdot 10^{-1}$
10/10/14	1.9279	6.2998	1.6375	$-3.1628 \cdot 10^{-1}$	1.9566	6.2760	$1.7006 \cdot 10^{-1}$	$6.4006 \cdot 10^{-2}$

Tabla VII.2.4: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando A_{13} y E_{13} son computados.

Por último hemos aplicado el Algoritmo Híbrido (AH) a la determinación de los valores de A_{13} y E_{13} correspondientes a nuevas experiencias cinéticas no isotérmicas, diferentes a las del proceso de *Entrenamiento*, con lo que se realiza la *Predicción* de 2 pares de valores de parámetros A_{13}/E_{13} . Los resultados obtenidos mediante la *Predicción* con ANN, son sometidos a un proceso posterior de refinamiento o mejora mediante la aplicación del algoritmo AGDC, obteniendo así los valores óptimos de A_{13}/E_{13} (Tabla VII.2.5). El proceso de optimización de A_{13}/E_{13} constituye el final del AH que nos proporciona los valores desconocidos de los

parámetros correspondientes a las cinéticas experimentales realizadas en el laboratorio y elegidas para llevar a cabo el proceso de *Predicción*.

(ANN		-	AGDC)	AH	
Valores Reales (A_{13})/ 10^{10} min^{-1}	Valores Reales (E_{13})/ 10^4 Jmol^{-1}	Valores Optimizados (A_{13})/ 10^{10} min^{-1}	Valores Optimizados (E_{13})/ 10^4 Jmol^{-1}	% Des. (A_{13})	% Des. (E_{13})
1.9589	6.2924	1.9590	6.2799	$5.2809 \cdot 10^{-2}$	$2.0997 \cdot 10^{-3}$
1.9252	6.0405	1.9554	6.0345	-1.5652	$9.4054 \cdot 10^{-2}$

Tabla VII.2.5: Resultados de los 2 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (25, 3, 2) con 3 *hidden layers* y Configuración 10/7/10, cuando A_{13}/E_{13} son computados después de la aplicación de AH (ANN-AGDC).

VII.2.2.2.- Determinación de ΔS_{12}^\ddagger y ΔH_{12}^\ddagger

VII.2.2.2.1.- Tratamiento de datos sintéticos

VII.2.2.2.1.a- *Predicción* de ΔS_{12}^\ddagger y ΔH_{12}^\ddagger mediante ANN

i) Diseño de Experimentos

En este caso, en el ED están implicados dos *factores*, correspondientes a ΔS_{12}^\ddagger y ΔH_{12}^\ddagger de la ecuación de Eyring, cuyas *respuestas* son los datos no isotérmicos cinéticos de la base de las curvas de *Inputs* (α_1/t). Los valores de los *factores* a partir de los cuales se realiza el correspondiente ED han sido adquiridos en la bibliografía: $\Delta S_{12}^\ddagger = -172 \text{ JK}^{-1} \text{ mol}^{-1}$ y $\Delta H_{12}^\ddagger = 40000 \text{ Jmol}^{-1}$. El ED que hemos seleccionado como óptimo es el denominado *Central Star Composite Experimental Design* (CSCED) cuya distribución se encuentra en la Figura VII.2.9.

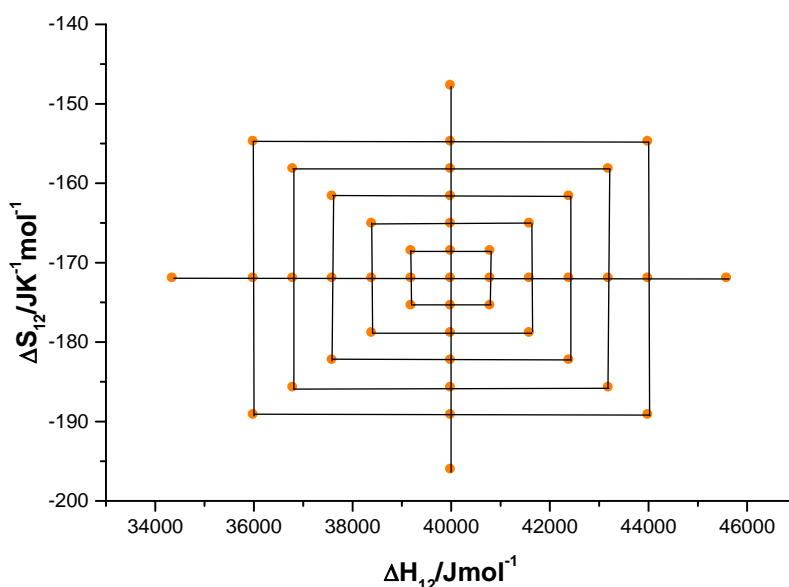


Figura VII.2.9: Diseño de Experimentos (CSCED) constituido por dos *factores* ($\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$) y 45 puntos distribuidos en 9 niveles (puntos 1-37) y 4 sub-niveles (puntos 38-45) tomando como base de curvas la matriz de *Input* usada para el proceso de *Entrenamiento* de la Red.

ii) Entrenamiento

Con los valores de ΔS_{12}^{\ddagger} y ΔH_{12}^{\ddagger} obtenidos en el ED se generan 45 curvas cinéticas no isotérmicas, cada una de ellas constituida por 50 pares de datos de fracción remanente (α_1/t) a las que se les impone un *noise* del orden del error experimental ($\pm 1.0 \cdot 10^{-4}$), considerando un hasta que la reacción ha llegado al 75% y la función T/t es de tipo polinómica de 5° grado. Una vez generadas las 45 curvas sintéticas, hemos llevado a cabo el proceso de *Entrenamiento* de la Red Neuronal Artificial (ANN) con el fin de encontrar la óptima que nos permita realizar la *Predicción* de ΔS_{12}^{\ddagger} y ΔH_{12}^{\ddagger} de manera satisfactoria. En todos los casos hemos utilizado el mismo porcentaje de curvas para *Training/Validation/Testing* (80/10/10) y el algoritmo *Backpropagation* (Levenberg-Marquardt). Hemos analizado varias situaciones, considerando en primer lugar una ANN cuya Arquitectura consta de 1 capa Oculta y en la que se analizan distintas Configuraciones que fue descartada debido a los malos resultados obtenidos. Posteriormente consideramos la ANN formada por 2 y 3 capas Ocultas analizando distintas Configuraciones y llegando a la conclusión que la más adecuada Arquitectura es la constituida por 2 capas y Configuración 10/10. Con objeto de confirmar que esa es la Configuración óptima hemos evaluado los resultados teniendo en cuenta el valor de MSE ($\approx 10^{-9}$), las Regresiones lineales *Outputs/Targets*, los perfiles de las curvas de *Trainig*, *Validation* y *Testing*, etc... La representación conjunta de las regresiones *Outputs /Targets* de los parámetros de ΔH_{12}^{\ddagger} y ΔS_{12}^{\ddagger} que presentan de nuevo órdenes de magnitud muy diferentes ($\Delta H_{12}^{\ddagger} \approx 10^4$ y $\Delta S_{12}^{\ddagger} \approx 10^2$) provoca que cada uno de los parámetros se encuentre en un extremo de la gráfica (Figura VII.2.10). Esto hace que la verdadera dispersión no pueda ser observada, siendo a simple vista aceptable, aunque ya se puede ver una nube de puntos en ΔH_{12}^{\ddagger} que no tienden a una recta.

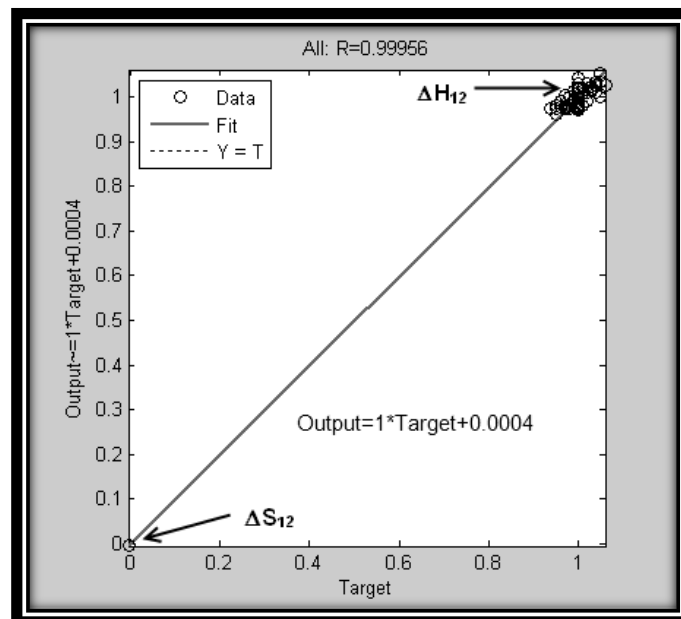


Figura VII.2.10: Regresión lineal de las matrices de *Output vs Target*, cuando ambos parámetros ($\Delta H_{12}^{\ddagger}/\Delta S_{12}^{\ddagger}$) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una arquitectura de 2 *hidden layers* (50, 2, 2).

Por el contrario, si tenemos en cuenta las regresiones *Outputs/Targets* por separado como en el caso de E_{12} y A_{12} , llegamos a la misma conclusión. Esto es, existe una gran dispersión en ambos parámetros (Figuras VII.2.11.a y b), que hace que los resultados de la aplicación de ANN sean inaceptables, lo que implica que sea necesario utilizar otro algoritmo, en nuestro caso AGDC para obtener unos valores finales de los PTA satisfactorios.

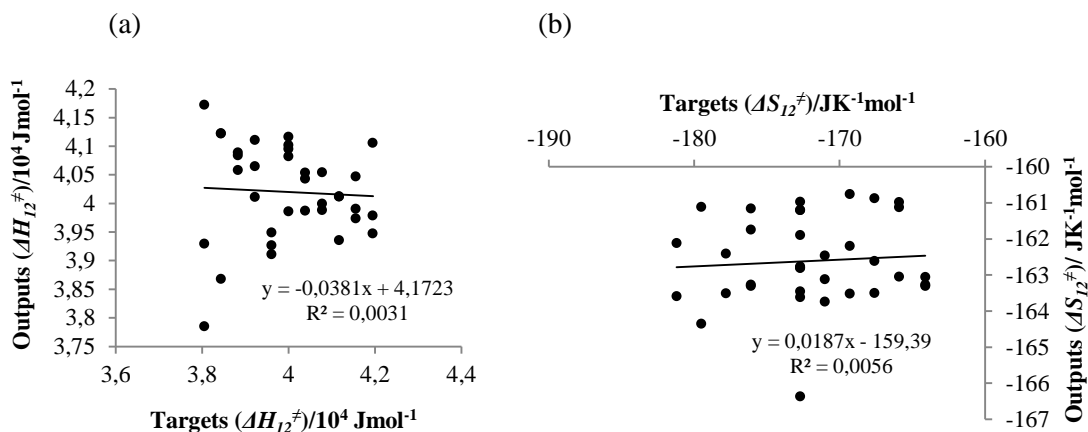


Figura VII.2.11 a y b: Regresión lineal de las matrices de *Outputs vs Targets*, exclusivamente en el caso de ΔH_{12}^\ddagger (a) y ΔS_{12}^\ddagger (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 2 *hidden layers* (50, 2, 2).

iii) Predicción

El proceso de *Entrenamiento* y posterior *Predicción* se ha realizado para un numeroso conjunto de Redes Neuronales considerando todas las Configuraciones posibles para las Arquitecturas con 2 y 3 capas Ocultas, descartando las constituidas por una sola capa. Se ha ensayado con un elevado número de casos, aunque en la Tabla VII.2.6 sólo mostramos los mejores resultados. Todos ellos fueron obtenidos para las Configuraciones más significativas correspondientes al punto central de las 45 curvas de ED. El par de valores de *Outputs* ($\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$) y las correspondientes desviaciones (“Des. %”) se muestran en la primera parte de la Tabla VII.2.6. Los valores obtenidos para los errores de los parámetros indican que se deben mejorar mediante la aplicación del algoritmo AGDC (segunda etapa del Algoritmo Híbrido).

VII.2.2.2.1.b- Optimización de ΔH_{12}^\ddagger y ΔS_{12}^\ddagger mediante AGDC

Los valores de $\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$ que han sido obtenidos mediante el proceso de *Predicción* realizado con ANN, son sometidos a un proceso de mejora y refinamiento mediante la aplicación del programa KINNOISOT(AGDC). La segunda parte de la Tabla VII.2.6 muestra los valores finales de los parámetros optimizados después de aplicar el algoritmo AGDC y las desviaciones de los valores reales (“Des. %”). La comparación de los de valores de las desviaciones obtenidas tras la aplicación de AGDC (columnas octava y novena) con los calculados a partir de ANN (columnas cuarta y quinta), observando que existe una gran mejora.

Config.	ANN				(ANN - AGDC) AH			
	$\Delta H_{12}^{\ddagger}/10^4$ Jmol ⁻¹	$\Delta S_{12}^{\ddagger}/10^2$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{12}^{\ddagger})	Des. % (ΔS_{12}^{\ddagger})	$\Delta H_{12}^{\ddagger}/10^4$ Jmol ⁻¹	$\Delta S_{12}^{\ddagger}/10^2$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{12}^{\ddagger})	Des. % (ΔS_{12}^{\ddagger})
8/10	3.8990	-1.6814	2.5253	2.2419	3.9964	-1.7202	9.0446.10 ⁻²	-1.4220.10 ⁻²
9/10	4.0398	-1.6424	-9.9468.10 ⁻¹	4.5081	3.9703	-1.7298	7.4833.10 ⁻¹	-5.7488.10 ⁻¹
10/10	4.0226	-1.7540	-5.6396.10 ⁻¹	-1.9806	3.9866	-1.7243	3.3666.10 ⁻¹	-2.5033.10 ⁻¹
12/10	4.0876	-1.9111	-2.1903	-11.115	4.0092	-1.7159	-2.3622.10 ⁻¹	2.4199.10 ⁻¹
10/9/10	3.9931	-1.7945	1.7251.10⁻¹	-4.3315	4.0007	-1.7199	2.1770.10⁻²	-3.5991.10⁻³
10/10/10	4.0532	-1.6116	-1.3301	6.3006	4.0304	-1.7094	-7.6244.10 ⁻¹	-6.1644.10 ⁻¹
10/10/11	4.0230	-1.7352	-5.7590.10 ⁻¹	-8.8660.10 ⁻¹	3.9957	-1.7207	1.1077.10 ⁻¹	-4.2770.10 ⁻²
12/10/10	4.0559	-1.6584	-1.3967	3.5804	3.9949	-1.7216	1.3999.10 ⁻¹	-9.4661.10 ⁻²

Tabla VII.2.6: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando ΔH_{12}^{\ddagger} y ΔS_{12}^{\ddagger} son computados.

Además los resultados de la Tabla VII.2.6 permiten llegar a la conclusión de que la Arquitectura idónea para llevar a cabo el proceso de *Predicción* es aquella con 3 capas y una Configuración 10/9/10 nodos respectivamente. Esta Arquitectura ha sido la que hemos utilizado para realizar los procesos posteriores de *Predicción* con ANN y Optimización mediante AGDC.

La mejora de los resultados cuando se aplica el AH es evidente y la podemos apreciar con claridad al realizar una representación conjunta (Figura VII.2.12) de los Residuales de α_1 calculados con los valores de los parámetros obtenidos de la aplicación de ANN y los conseguidos de la aplicación del Algoritmo Híbrido (AH).

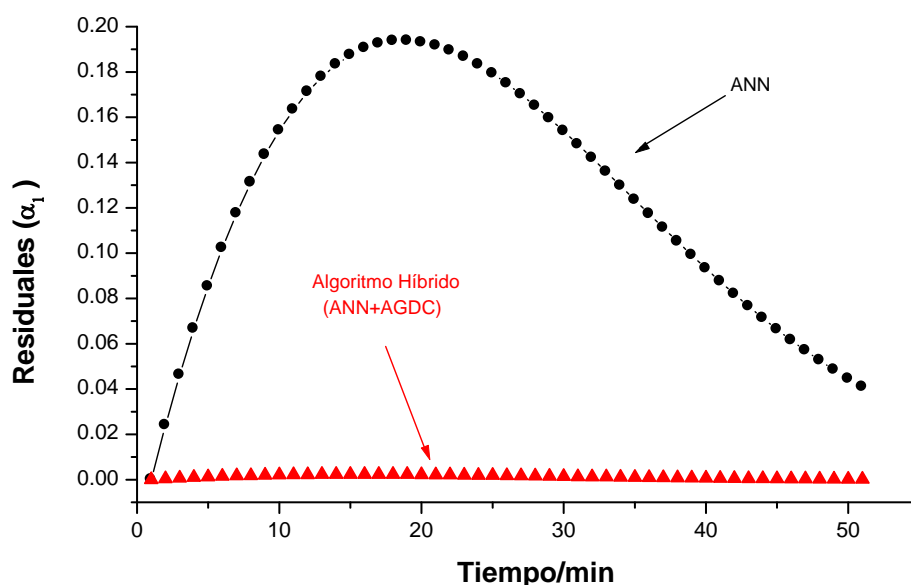


Figura VII.2.12: Figura de las residuales de α_2 calculados a partir de la ecuación [VII.2.1] con los valores de los parámetros ΔH_{12}^{\ddagger} y ΔS_{12}^{\ddagger} obtenidos mediante la aplicación de la ANN y los obtenidos a partir de la ecuación [VII.2.2] después de la aplicación del algoritmo completo.

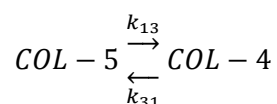
Por último hemos aplicado el Algoritmo Híbrido (AH) a la determinación de los valores de $\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$ correspondientes a 8 nuevas experiencias cinéticas no isotérmicas, generadas a partir de pares de valores comprendidos dentro de los intervalos del ED. El proceso de *Predicción* se llevó a cabo con la Red Neuronal óptima y posteriormente se realiza la optimización mediante la aplicación del algoritmo AGDC. El proceso constituye el final del AH que nos proporciona los valores desconocidos de los parámetros correspondientes a las experiencias cinéticas elegidas para llevar a cabo el proceso de *Predicción*. Los resultados obtenidos para los pares de valores $\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$ y sus correspondientes desviaciones (“% Des.”) se muestran en la Tabla VII.2.7.

(ANN		-	AGDC)	AH	
Valores Reales (ΔH_{12}^\ddagger)/10 ⁴ Jmol ⁻¹	Valores Reales (ΔS_{12}^\ddagger)/10 ² JK ⁻¹ mol ⁻¹	Valores Optimizados (ΔH_{12}^\ddagger)/10 ⁴ Jmol ⁻¹	Valores Optimizados (ΔS_{12}^\ddagger)/10 ² JK ⁻¹ mol ⁻¹	% Des. (ΔH_{12}^\ddagger)	% Des. (ΔS_{12}^\ddagger)
4.0289	-1.7671	4.0214	-1.4.497	4.3944.10 ⁻¹	-3.3177.10 ⁻¹
4.2358	-1.8018	4.1366	-1.7864	4.8663.10 ⁻³	3.6468.10 ⁻³
4.0121	-1.7494	3.9553	-1.7458	1.4133.10 ⁻¹	-1.0488.10 ⁻¹
4.0180	-1.7560	3.9843	-1.7468	2.2030	3.1080
4.0016	-1.7259	3.9403	-1.7024	3.6334.10 ⁻²	-5.9442.10 ⁻²
3.9770	-1.7311	3.8591	-1.6679	1.1011.10 ⁻¹	-3.1997.10 ⁻²
4.0074	-1.7439	4.0654	-1.6994	-1.7444.10 ⁻¹	1.3944.10 ⁻¹
4.0046	-1.7400	4.0373	-1.7000	2.4630	-1.9000

Tabla VII.2.7: Resultados de los 8 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (50, 3, 2) con 3 *hidden layers* y Configuración 10/9/10, cuando $\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$ son computados después de la aplicación de AH (ANN-AGDC).

VII.2.2.2.2.-Tratamiento de datos experimentales

Una vez comprobada la aplicabilidad del AH para la determinación de ΔH_{12}^\ddagger y ΔS_{12}^\ddagger en el caso del mecanismo de reacción $B_1 \xrightarrow{k_{12}} B_2$, hemos aplicado el AH para la determinación de estos parámetros correspondientes a la reacción de isomerización del esteroide colest-5-en-3-ona, catalizada por etóxido sódico en medio etanol. Tal y como hemos visto en el apartado VII.2.2.1.2 la reacción global de isomerización puede escribir como:



En este caso aplicamos el AH para el estudio de la reacción en el sentido directo, hemos realizado un estudio de la cinética no isotérmica tal y como se explicó en el apartado VII.2.2.1.2.

VII.2.2.2.2.a- Predicción de ΔH_{13}^\ddagger y ΔS_{13}^\ddagger mediante ANN

i) Diseño de Experimentos

El ED más idóneo obviamente es del tipo *Central Star Composite Experimental Design* (CSCED), cuyos *factores* están formados por el conjunto de parámetros $\Delta H_{13}^\ddagger/\Delta S_{13}^\ddagger$ y las *respuestas* son los datos de la base de curvas de *Inputs* (α_1/t y α_3/t). Los valores centrales de ambos *factores* han sido tomados de bibliografía^(70-71,157) y son los siguientes: $\Delta H_{13}^\ddagger = 6.03 \cdot 10^4$ Jmol^{-1} y $\Delta S_{13}^\ddagger = -90.5 \text{ JK}^{-1}\text{mol}^{-1}$. El Diseño de Experimentos (ED) obtenido para este caso se muestra en la Figura VII.2.13.

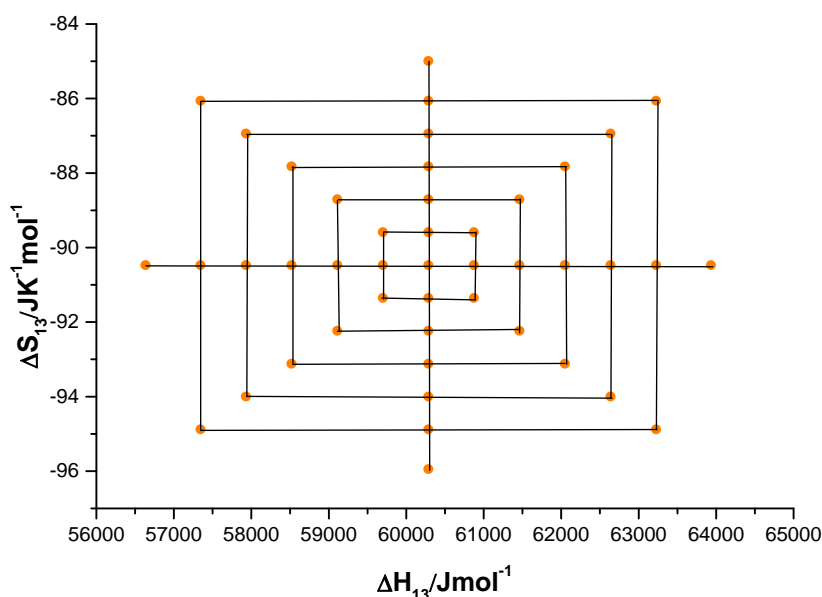


Figura VII.2.13: Diseño de Experimentos (CSCED) constituido por dos *factores* ($\Delta H_{13}^\ddagger/\Delta S_{13}^\ddagger$) y 45 puntos distribuidos en 9 niveles (puntos 1-37) y 4 sub-niveles (puntos 38-45) tomando como base de curvas la matriz de *Input* usada para el proceso de *Entrenamiento* de la Red.

ii) Entrenamiento

A partir del ED realizado, se han generado las curvas experimentales no isotérmicas (*Inputs*) considerando una variación de la temperatura hiperbólica y se ha llevado a cabo el proceso de *Entrenamiento* de la ANN con el fin de encontrar la más adecuada para realizar el proceso de *Predicción* de ΔH_{13}^\ddagger y ΔS_{13}^\ddagger . Para determinar la Arquitectura óptima de la Red Neuronal se realizó el *Entrenamiento* variando el número de nodos y la Configuración de las capas ocultas sistemáticamente, considerando una amplia casuística de las opciones posibles y utilizando en todos los casos el mismo porcentaje de curvas para el *Entrenamiento*, *Validación* y *Test* (80/10/10). Tras la experimentación comprobamos que la Red Neuronal idónea consta de una Estructura óptima (25, 3, 2) y Configuración 12/10/10.

En la representación conjunta de las regresiones *Outputs/Targets* de ΔH_{13}^\ddagger y ΔS_{13}^\ddagger (Figura VII.2.14) se observa que se encuentran muy separadas dentro de la gráfica puesto que

sus órdenes de magnitud son muy diferentes ($\Delta H_{13}^{\ddagger} \approx 10^4$ y $\Delta S_{13}^{\ddagger} \approx 10^1$) y al estar representados en la misma escala, los puntos que aparecen en el gráfico forman dos grupos separados por una gran distancia. Además todos los valores correspondientes a ΔS_{13}^{\ddagger} se superponen en el mismo punto. Esta situación enmascara los efectos reales de la dispersión, lo que provoca que la recta de regresión cuando ambos PTA están representados conjuntamente sea correcta.

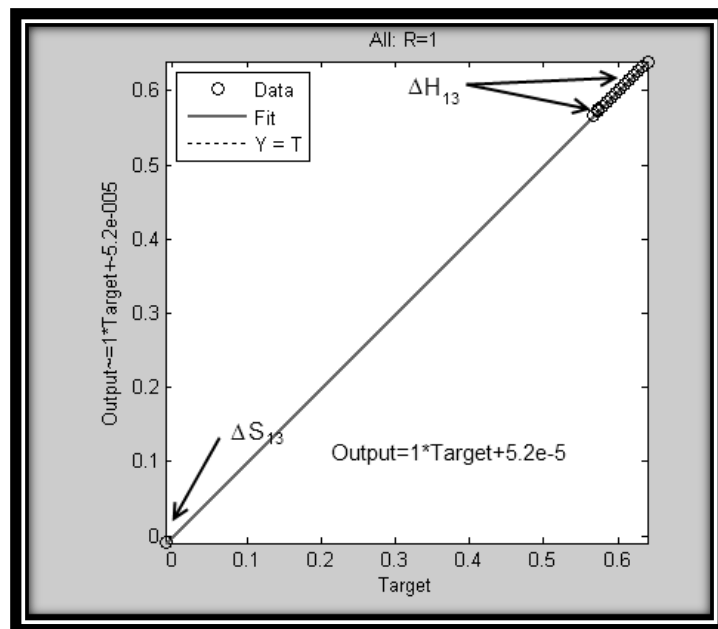


Figura VII.2.14: Regresión lineal de las matrices de *Outputs vs Targets*, cuando ambos parámetros ($\Delta S_{13}^{\ddagger}/\Delta H_{13}^{\ddagger}$) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* (25, 3, 2).

En el caso de realizar las representaciones de las regresiones *Outputs/Targets* de ΔH_{13}^{\ddagger} y ΔS_{13}^{\ddagger} (Figuras VII.2.15.a y b), de forma individual se observa una gran dispersión en ambos parámetros. Esta característica provoca que los resultados obtenidos mediante la aplicación exclusiva de ANN, no sean aceptables. Por este motivo es necesario utilizar otro algoritmo, en nuestro caso AGDC, para obtener valores finales satisfactorios de los parámetros.

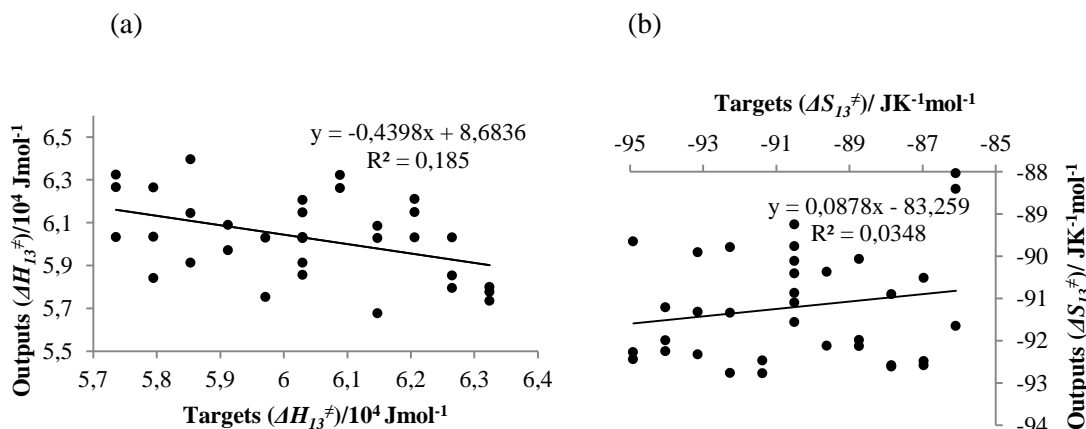


Figura VII.2.15.a y b: Regresión lineal de las matrices de *Outputs vs Targets*, exclusivamente en el caso de ΔH_{13}^{\ddagger} (a) y ΔS_{13}^{\ddagger} (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* (25, 3, 2).

iii) *Predicción*

Para llevar a cabo el proceso de *Predicción* de ΔH_{13}^{\ddagger} y ΔS_{13}^{\ddagger} en el caso de datos cinéticos experimentales procedentes de experiencias cinéticas en las que su valor es desconocido, hemos utilizado la ANN óptima. Hemos realizado el proceso de *Entrenamiento* de un conjunto de ANN considerando todas las Configuraciones posibles para las Arquitecturas con 2 y 3 capas Ocultas. En la Tabla VII.2.8 se muestran los resultados más relevantes, obtenidos para las Configuraciones más significativas correspondientes al punto central de las 45 curvas de ED. En la primera parte de la Tabla VII.2.8 se muestran los valores de *Outputs*, $\Delta H_{13}^{\ddagger}/\Delta S_{13}^{\ddagger}$ y las correspondientes desviaciones (“Des. %”). Los valores obtenidos para los errores de los parámetros indican que es necesario mejorar los resultados mediante la aplicación del AGDC.

VII.2.2.2.b- Optimización de ΔS_{13}^{\ddagger} y ΔH_{13}^{\ddagger} mediante AGDC

Con objeto de mejorar los valores de ΔH_{13}^{\ddagger} y ΔS_{13}^{\ddagger} obtenidos en el proceso de *Predicción* realizado con ANN, se optimizaron mediante la aplicación del programa **KINNOISOT(AGDC)** utilizando como estimaciones iniciales los valores predichos mediante ANN. La segunda parte de la Tabla VII.2.8 muestra los valores de ΔH_{13}^{\ddagger} y ΔS_{13}^{\ddagger} optimizados con el algoritmo AGDC y las desviaciones de los valores reales (“Des. %”). Comparando los valores de las desviaciones (columnas octava y novena) de ΔH_{13}^{\ddagger} y ΔS_{13}^{\ddagger} con los calculados por la aplicación ANN (columnas cuarta y quinta) se observa una gran mejora. Los resultados que se ofrecen en la Tabla VII.2.8 nos permiten concluir que la mejor Arquitectura para llevar a cabo el proceso de *Predicción* es aquella que consta de 3 capas y una Configuración 12/10/10.

Config.	ANN				(ANN - AGDC) AH			
	$\Delta H_{I3}^{\#}/10^4$ Jmol ⁻¹	$\Delta S_{I3}^{\#}/$ JK ⁻¹ mol ⁻¹	Des. % ($\Delta H_{I3}^{\#}$)	Des. % ($\Delta S_{I3}^{\#}$)	$\Delta H_{I3}^{\#}/10^4$ Jmol ⁻¹	$\Delta S_{I3}^{\#}/$ JK ⁻¹ mol ⁻¹	Des. % ($\Delta H_{I3}^{\#}$)	Des. % ($\Delta S_{I3}^{\#}$)
8/10	6.0308	-90.049	-1.3808.10 ⁻²	4.9839.10 ⁻¹	6.0318	-90.443	-2.9118.10 ⁻²	6.2643.10 ⁻²
9/10	6.0371	-91.375	-1.1871.10 ⁻¹	-9.6635.10 ⁻¹	6.0318	-90.399	-4.6118.10 ⁻²	1.1219.10 ⁻¹
10/8	6.0318	-94.818	-2.9918.10 ⁻²	-4.7711	6.0392	-90.128	-1.5242.10 ⁻¹	4.1118.10 ⁻¹
10/9	6.0436	-87.700	-2.2646.10 ⁻¹	3.0939	6.0178	-90.907	2.0278.10 ⁻¹	-4.4977.10 ⁻¹
10/10	6.0322	-96.858	-3.7422.10 ⁻²	-7.0258	5.9727	-92.583	9.5007.10 ⁻¹	-2.3011
10/11	6.0375	-90.570	-1.2585.10 ⁻¹	-7.7870.10 ⁻²	6.0327	-90.409	-4.5427.10 ⁻²	1.0079.10 ⁻¹
10/12	6.0270	-80.201	4.8370.10 ⁻²	11.380	6.0283	-90.365	2.8983.10 ⁻²	1.4905.10 ⁻¹
11/10	6.0310	-91.517	-1.7310.10 ⁻²	-1.1239	6.0271	-90.592	4.8071.10 ⁻²	-1.0122.10 ⁻¹
12/10	6.0036	-89.777	4.3616.10 ⁻¹	7.9897.10 ⁻¹	6.0275	-90.578	4.0875.10 ⁻²	-8.6178.10 ⁻²
8/10/10	6.0307	-91.991	-1.2407.10 ⁻²	-1.6477	6.0291	-90.529	1.5591.10 ⁻²	-3.2229.10 ⁻²
9/10/10	6.0360	-94.019	-1.0050.10 ⁻¹	-3.8886	6.0161	-90.896	2.3041.10 ⁻¹	-4.3756.10 ⁻¹
10/8/10	6.0271	-91.831	4.7171.10 ⁻²	-1.4708	6.0314	-90.451	-2.2814.10 ⁻²	5.4051.10 ⁻²
10/9/10	6.0362	-95.052	-1.0362.10 ⁻¹	-5.0296	5.9959	-91.521	5.6519.10 ⁻¹	-1.1282
10/10/8	6.0345	-86.522	-7.5845.10 ⁻²	4.3954	6.0425	-90.095	2.0665.10 ⁻¹	4.5145.10 ⁻¹
10/10/9	6.0412	-91.534	-1.8722.10 ⁻¹	-1.1423	6.0352	-90.312	8.6752.10 ⁻²	2.0742.10 ⁻¹
10/10/10	6.0336	-92.451	-6.1136.10 ⁻²	-2.1556	6.0345	-90.333	-7.4245.10 ⁻²	1.8513.10 ⁻¹
10/10/11	6.0411	-93.390	-1.8491.10 ⁻¹	-3.1934	6.0286	-90.495	2.4086.10 ⁻²	5.7495.10 ⁻³
10/10/12	6.0320	-89.701	-3.4120.10 ⁻²	8.8271.10 ⁻¹	6.0335	-90.386	-5.7935.10 ⁻²	1.2636.10 ⁻¹
10/11/10	5.9612	-96.321	1.1403	-6.4318	6.0440	-90.066	-2.3260.10 ⁻¹	4.4926.10 ⁻¹
10/12/10	6.0367	-91.272	-1.1147.10 ⁻¹	-8.5312.10 ⁻¹	6.0330	-90.391	-5.0530.10 ⁻²	1.2081.10 ⁻¹
11/10/10	6.0204	-90.563	0.15834	-6.9063.10 ⁻²	6.0245	-90.692	9.4045.10 ⁻²	-2.1232.10 ⁻¹
12/10/10	6.0294	-89.790	0.0097294	7.8450.10⁻¹	6.0308	-90.476	1.3808.10⁻²	2.6776.10⁻²

Tabla VII.2.8: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando $\Delta S_{I3}^{\#}$ y $\Delta H_{I3}^{\#}$ son computados.

La mejora de los resultados cuando se aplica el AH es evidente y se aprecia con claridad al realizar una representación conjunta (Figura VII.2.16) de los Residuales de α_2 calculados con los valores de los parámetros obtenidos de la aplicación de ANN y los conseguidos de la aplicación del Algoritmo Híbrido (AH), calculados a partir de las siguientes fórmulas [VII.2.1] y [VII.2.2].

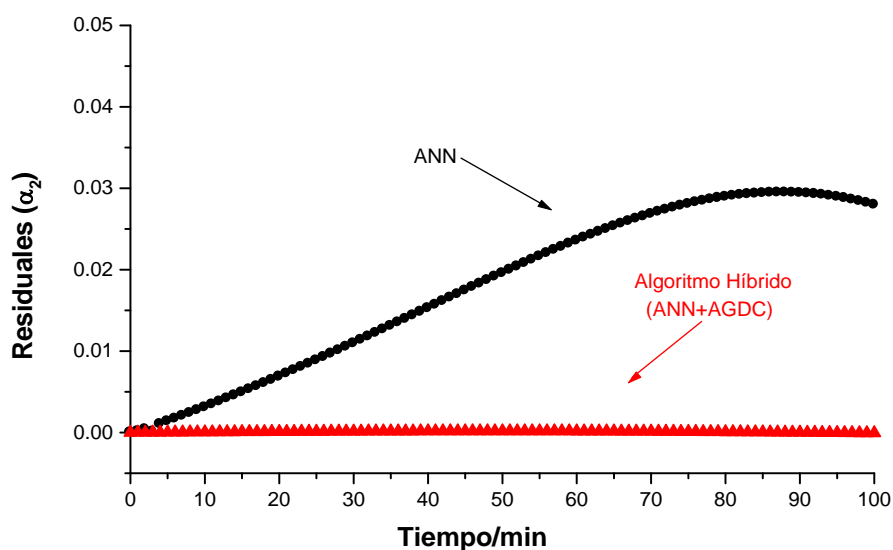


Figura VII.2.16: Figura de las residuales de α_2 calculados a partir de la ecuación [VII.2.1] con los valores de los parámetros ΔH_{13}^\ddagger y ΔS_{13}^\ddagger obtenidos mediante la aplicación de la ANN y los obtenidos a partir de la ecuación [VII.2.2] después de la aplicación del algoritmo completo.

Por último hemos aplicado el AH para determinar los valores de ΔH_{13}^\ddagger y ΔS_{13}^\ddagger correspondientes a nuevas experiencias cinéticas no isotérmicas. Inicialmente se realizó el proceso de *Predicción* con la ANN óptima. Posteriormente los resultados obtenidos son sometidos a un proceso posterior de optimización mediante la aplicación del algoritmo AGDC, obteniendo así los valores idóneos de ΔH_{13}^\ddagger y ΔS_{13}^\ddagger . Los resultados obtenidos para los pares de valores $\Delta H_{13}^\ddagger/\Delta S_{13}^\ddagger$ se muestran en la Tabla VII.2.9.

(ANN		-	AGDC)	AH	
Valores Reales (ΔH_{13}^\ddagger)/ 10^4 Jmol ⁻¹	Valores Reales (ΔS_{13}^\ddagger)/JK ⁻¹ mol ⁻¹	Valores Optimizados (ΔH_{13}^\ddagger)/ 10^4 Jmol ⁻¹	Valores Optimizados (ΔS_{13}^\ddagger)/JK ⁻¹ mol ⁻¹	% Des. (ΔH_{13}^\ddagger)	% Des. (ΔS_{13}^\ddagger)
5.8539	-92.254	4.0214	-93.151	$1.2214 \cdot 10^{-3}$	$-4.2151 \cdot 10^{-3}$
6.2106	-92.616	4.1366	-91.473	$1.5856 \cdot 10^{-1}$	$-3.5703 \cdot 10^{-1}$
5.9122	-91.320	3.9553	-92.272	$2.2553 \cdot 10^{-3}$	$-7.4272 \cdot 10^{-3}$
6.1491	-92.334	3.9843	-92.254	$-5.4843 \cdot 10^{-3}$	$1.1854 \cdot 10^{-2}$
6.0852	-90.069	3.9403	-89.712	$4.6303 \cdot 10^{-2}$	$-1.0512 \cdot 10^{-1}$
6.0891	-91.342	3.8591	-91.370	$-6.9591 \cdot 10^{-3}$	$1.4370 \cdot 10^{-2}$
5.7530	-90.371	4.0373	-90.504	$1.9373 \cdot 10^{-3}$	$-4.7504 \cdot 10^{-3}$

Tabla VII.2.9: Resultados de los 7 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (25, 3, 2) con 3 *hidden layers* y Configuración 12/10/10, cuando $\Delta S_{13}^\ddagger/\Delta H_{13}^\ddagger$ son computados después de la aplicación de AH (ANN-AGDC).

VII.2.3.- Mecanismo II. $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$

La determinación de los Parámetros Termodinámicos de Activación correspondientes a la primera etapa de este mecanismo se realiza tal y como se ha indicado en los apartados VII.2.2.1.1 (E_{12} y A_{12}) y VII.2.2.2.1 (ΔS_{12}^\ddagger y ΔH_{12}^\ddagger) por lo que nos centraremos en la determinación de los PTA correspondientes a la segunda etapa del mecanismo (E_{23} y A_{23}) (apartado VII.2.3.1) y ΔS_{23}^\ddagger y ΔH_{23}^\ddagger (apartado VII.2.3.2).

VII.2.3.1.-Determinación de E_{23} y A_{23}

VII.2.3.1.1.-Tratamiento de datos sintéticos

VII.2.3.1.1.a-Predicción de E_{23} y A_{23} mediante ANN

i) Diseño de Experimentos

El Diseño de Experimentos (ED) que hemos elegido como el más idóneo es el tipo *Central Star Composite Experimental Design* (CSCED) y en este caso los *factores* son el conjunto de parámetros A_{23}/E_{23} para la ecuación de Arrhenius y las *respuestas* en las cinéticas no isotérmicas son los datos de la base de curvas de *Input* (α_2/t , y α_3/t). Los valores para A_{23}/E_{23} con los que se generarán los datos sintéticos para el mecanismo que estamos considerando, han sido tomados de bibliografía y son los siguientes: $E_{23} = 4.25 \cdot 10^4 \text{ Jmol}^{-1}$ y $A_{23} = 5 \cdot 10^5 \text{ min}^{-1}$. De acuerdo con los resultados del estudio de las funciones $T=f(t)$, hemos generado los datos de las cinéticas no isotérmicas para la ecuación de Arrhenius con el fin de obtener las curvas de *Inputs* para realizar el proceso de *Entrenamiento* de la Red, con los PTA organizados de acuerdo con el Diseño de Experimentos (ED) (Figura VII.2.17).

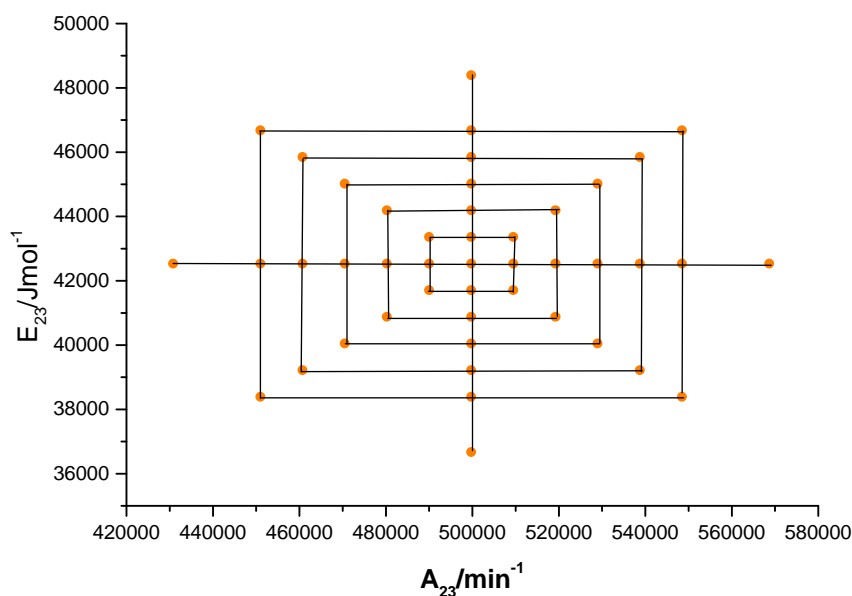


Figura VII.2.17: Diseño de Experimentos (CSCED) constituido por dos *factores* (A_{23}/E_{23}) y 45 puntos distribuidos en 9 niveles (puntos 1-37) y 4 sub-niveles (puntos 38-45) tomando como base de curvas la matriz de *Input* usada para el proceso de *Entrenamiento* de la Red.

ii) Entrenamiento

Una vez realizado el correspondiente Diseño de Experimentos se generan las matrices de *Inputs* formados por una base de 45 curvas cinéticas no isotérmicas con 100 pares de datos cinéticos (α_2/t , y α_3/t , siendo α_2 y α_3 las fracciones molares de las especies B_2 y B_3). En este caso el valor del *noise* es del orden del error experimental ($\pm 1.0 \cdot 10^{-4}$), el intervalo de tiempo buscado es aquel en el que la reacción ha llegado al 75% y la función T/t del tipo polinómica. Para determinar la arquitectura óptima de la Red Neuronal se realizó el *Entrenamiento* teniendo en cuenta las siguientes consideraciones: Algoritmo *Backpropagation* (Levenberg-Marquardt); porcentajes *Training/Validation/Testing* 80/10/10; consideramos una variación sistemática del número de capas Ocultas (*Hidden*) y sus Configuraciones (número de nodos). Teniendo en cuenta que la Configuración óptima para realizar la *Predicción* es aquella que posea el menor valor de MSE ($\approx 10^{-5}$), comprobamos que la Red Neuronal más apta consta de 2 capas (Estructura) y dentro de esa capa los nodos óptimos son 20/10 (Configuración).

En la evaluación de los resultados obtenidos en el proceso de *Entrenamiento* hemos observado la existencia de una ambigüedad importante que merece un comentario detallado para evitar obtener conclusiones erróneas. La Figura VII.2.18 muestra la representación gráfica de los resultados de los procesos de *Training*, *Validation* y *Testing* (“*ALL*”) para una Red con una Estructura óptima (100, 2, 2) y Configuración (20/10) cuando se determina A_{23} y E_{23} . La línea de Regresión obtenida de esta gráfica, en la que se representan simultáneamente ambos parámetros, tiene un valor de $R= 0.99999$, que corresponde a una buena linealidad entre los elementos de las matrices *Outputs/Targets*. Inicialmente estos resultados son aceptables, sin embargo, es necesario tener en cuenta que el orden de magnitud de los parámetros es muy diferente ($A_{23} \approx 10^5$ y $E_{23} \approx 10^4$) y están representados en la misma escala, por lo que los puntos aparecen en el gráfico formando dos grupos separados por una gran distancia entre ellos. Además, todos los valores correspondientes a E_{23} se superponen en el mismo punto. Esta situación enmascara los efectos reales de la dispersión, provocando que la regresión *Outputs/Targets* de los parámetros se ajuste a una línea recta cuando ambos PTA están representados juntos en el mismo gráfico. Como ya hemos comentado, este mismo comportamiento ha sido detectado en el sistema cinético anterior ($B_1 \xrightarrow{k_{12}} B_2$) y en éste se repite. Por tanto, va a resultar crucial para realizar el tratamiento correcto, aplicar de nuevo el Algoritmo Híbrido (ANN-AGDC) que proponemos en el presente trabajo.

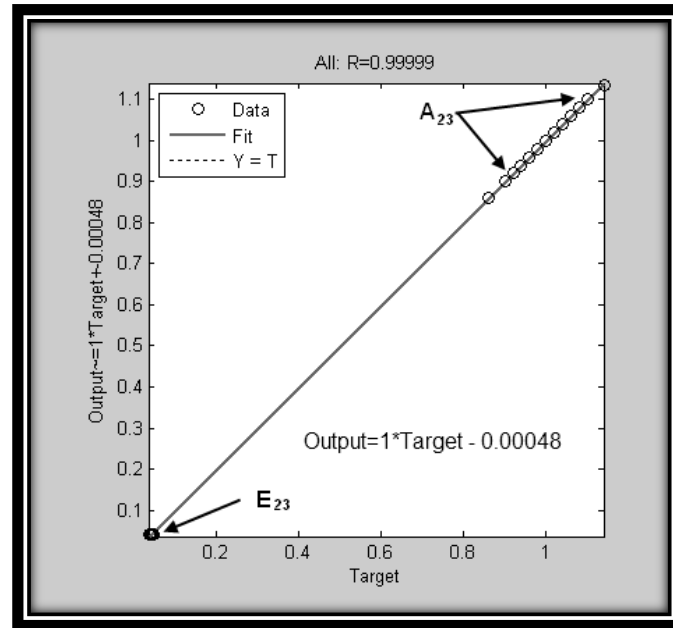


Figura VII.2.18: Regresión lineal de las matrices de *Outputs vs Targets*, cuando ambos parámetros (A_{23} y E_{23}) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 2 *hidden layers* (100, 2, 2).

Sin embargo, si tenemos en cuenta las regresiones lineales *Outputs/Targets* de cada parámetro por separado utilizando la escala apropiada, se detecta bastante dispersión para ambos parámetros (Figuras VII.2.19.a y b). Los valores obtenidos tanto para la ordenada en el origen (muy diferente de cero) como para R^2 , manifiestan claramente una mala correlación entre *Output/Target*, que hace inaceptables los valores de A_{23} y E_{23} determinados mediante ANN.

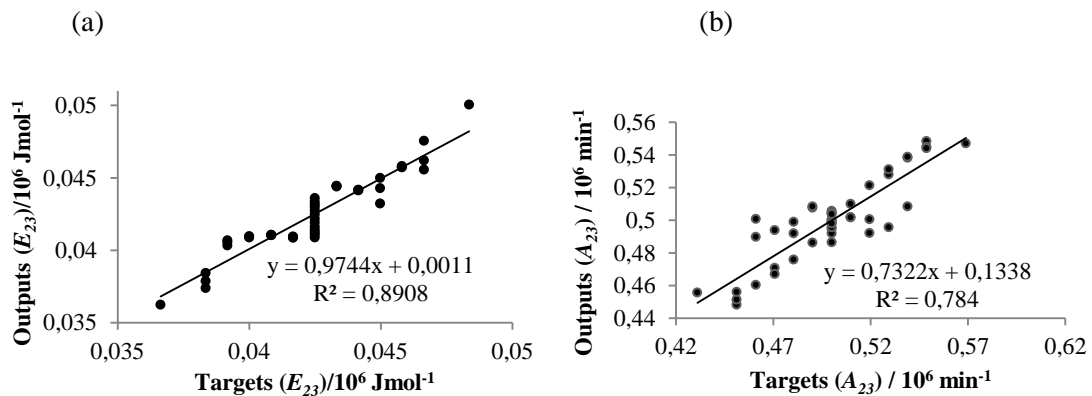


Figura VII.2.19.a y b: Regresión lineal de las matrices de *Output vs Target*, exclusivamente en el caso de E_{23} (a) y A_{23} (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 2 *hidden layers* (100, 2, 2).

En conclusión, estos valores son inaceptables, por lo que se requiere la aplicación completa del AH para mejorar los resultados. Los valores obtenidos para los PTA después de la aplicación del método ANN, serán las estimaciones iniciales para el algoritmo de gradiente AGDC, que nos proporcionará valores muy aceptables para los parámetros optimizados.

iii) *Predicción*

El proceso de *Predicción* consiste en la aplicación de la Red Neuronal de Arquitectura óptima a las nuevas curvas cinéticas, distintas a las consideradas en el proceso de *Entrenamiento*, y cuyos valores de los parámetros son desconocidos. La *Predicción* constituye el proceso final de la aplicación de la metodología ANN para determinar los parámetros desconocidos, que corresponde a cinéticas experimentales llevadas en el laboratorio.

Hemos llevado a cabo el proceso de *Entrenamiento* de un gran conjunto de Redes Neuronales, considerando todas las Configuraciones posibles para las Arquitecturas con 2 y 3 capas Ocultas, descartando “*a priori*” aquellas con una sola capa. En la Tabla VII.2.10 mostramos los resultados más relevantes. Todos ellos fueron obtenidos para las Configuraciones más significativas correspondientes al punto central de las 45 curvas de ED. El par de valores de *Outputs*, A_{23}/E_{23} , y las correspondientes desviaciones (“*Des. %*”) se muestran en la primera parte de la Tabla VII.2.10. Además, la segunda parte muestra los valores de los parámetros optimizados después de aplicar AGDC y las desviaciones de los valores reales (“*Des. %*”). Los valores obtenidos para los errores de los parámetros indican que es necesario mejorar los resultados mediante la aplicación de la segunda etapa del Algoritmo Híbrido (AH) que constituye la aplicación de AGDC.

VII.2.3.1.1.b- Optimización de E_{23} y A_{23} mediante AGDC

Los valores de E_{23} y A_{23} que han sido obtenidos mediante el proceso de *Predicción* realizado con ANN, son sometidos a un proceso de mejora y refinamiento mediante la aplicación del programa **KINNOISOT(AGDC)**. La Tabla VII.2.10 los valores de E_{23} y A_{23} resultantes de la aplicación de ANN y los obtenidos tras la aplicación del algoritmo AGDC. Se indican únicamente las configuraciones más significativas que nos proporcionan los mejores resultados y los obtenidos para la curva correspondiente al punto central de las 45 que componen el ED por ser el más importante. En la primera parte de la Tabla VII.2.10 se muestran los valores de *outputs* de E_{23} y A_{23} obtenidos con ANN y las desviaciones en % (“*Des. %*”) de estos valores con respecto a los que sirvieron para generar los datos. En la 2ª parte aparecen los valores de E_{23} y A_{23} optimizados con AGDC utilizando como estimaciones iniciales los valores procedentes del proceso de *Predicción* con ANN, además se muestran las desviaciones en % de los valores reales (“*Des. %*”). Después de la aplicación de AH observamos una evidente mejora al comparar los valores de las desviaciones (columnas octava y novena) de ambos parámetros con los calculados por la aplicación ANN (columnas cuarta y quinta).

Config.	ANN				(ANN - AGDC)			AH
	$A_{23}/10^5$ min^{-1}	$E_{23}/10^4$ Jmol^{-1}	Des. % (A_{23})	Des. % (E_{23})	$A_{23}/10^5$ min^{-1}	$E_{23}/10^4$ Jmol^{-1}	Des. % (A_{23})	Des. % (E_{23})
10/10	4.9626	4.2496	$7.4666 \cdot 10^{-1}$	$7.1496 \cdot 10^{-3}$	4.9740	4.2486	$5.2070 \cdot 10^{-1}$	$3.1886 \cdot 10^{-2}$
10/11	5.0512	4.2622	-1.0254	$-2.8912 \cdot 10^{-1}$	5.1192	4.2561	-2.3840	$-1.4391 \cdot 10^{-1}$
10/12	4.9752	4.3018	$4.9592 \cdot 10^{-1}$	-1.2200	5.0831	4.2543	-1.6619	$-1.0033 \cdot 10^{-1}$
10/14	4.9690	4.2858	$6.1900 \cdot 10^{-1}$	$-8.4338 \cdot 10^{-1}$	5.1641	4.2584	-3.2818	$-1.9664 \cdot 10^{-1}$
13/10	5.0672	4.2647	-1.3457	$-3.4727 \cdot 10^{-1}$	5.1469	4.2575	-2.9384	$-1.7695 \cdot 10^{-1}$
15/10	4.9809	4.2945	$3.8149 \cdot 10^{-1}$	-1.0470	5.1079	4.2356	-2.1979	$-1.3266 \cdot 10^{-1}$
20/10	5.0490	4.3250	$-9.8130 \cdot 10^{-1}$	-1.7665	5.0012	4.2501	$-2.3512 \cdot 10^{-1}$	$-1.6501 \cdot 10^{-3}$
7/10/10	5.0028	4.3079	$-5.7628 \cdot 10^{-2}$	-1.3625	5.0838	4.2543	-1.6762	$-1.0173 \cdot 10^{-1}$
9/10/10	5.1849	4.1884	-3.6988	1.4479	4.6510	4.2313	6.9801	$4.3993 \cdot 10^{-1}$

Tabla VII.2.10: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando E_{23} y A_{23} son computados.

La Arquitectura óptima de la Red Neuronal está constituida por 2 capas Ocultas (*Hidden*) con una configuración 20/10. Esta Arquitectura ha sido la que hemos utilizado para realizar los procesos posteriores de *Predicción* con ANN y Optimización mediante AGDC.

Por último hemos aplicado el Algoritmo Híbrido (AH) para determinar los valores de los parámetros E_{23} y A_{23} correspondientes a nuevas experiencias cinéticas no isotérmicas, no consideradas en el *Entrenamiento* correspondientes a pares de valores comprendidos dentro de los intervalos del ED. El proceso de *Predicción* se llevó a cabo con la Red Neuronal óptima para cuatro curvas cinéticas no isotérmicas. Los resultados obtenidos para la *Predicción* de los valores E_{23}/A_{23} mediante ANN, son sometidos a un proceso posterior de optimización mediante la aplicación del algoritmo AGDC, teniendo en cuenta que los porcentajes de desviaciones (“% *Des.*”) poseen una magnitud similar a la de las desviaciones del proceso de *Entrenamiento* de la Red Neuronal óptima. De esta manera podemos aceptar los valores optimizados de E_{23} y A_{23} como definitivos para los 4 procesos de *Predicción*. Los resultados obtenidos se muestran en la Tabla VII.2.11.

(ANN		- AGDC)		AH	
Valores Reales (A_{23})/ 10^5 min^{-1}	Valores Reales (E_{23})/ 10^4 Jmol^{-1}	Valores Optimizados (A_{23})/ 10^5 min^{-1}	Valores Optimizados (E_{23})/ 10^4 Jmol^{-1}	% Des. (A_{23})	% Des. (E_{23})
5.0830	4.0306	5.7654	5.3756	30.062	-17.326
4.9202	4.1031	3.6072	4.1655	27.856	1.9876
5.0173	4.0928	5.0621	4.0925	-3.2565	5.5473
4.9789	4.2181	5.0090	3.9874	5.3773	$3.4994 \cdot 10^{-1}$

Tabla VII.2.11: Resultados de los 4 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (100, 2, 2) con 2 *hidden layers* y Configuración 20/10, cuando E_{23} y A_{23} son computados después de la aplicación de AH (ANN-AGDC).

VII.2.3.1.2.- Tratamiento de datos experimentales

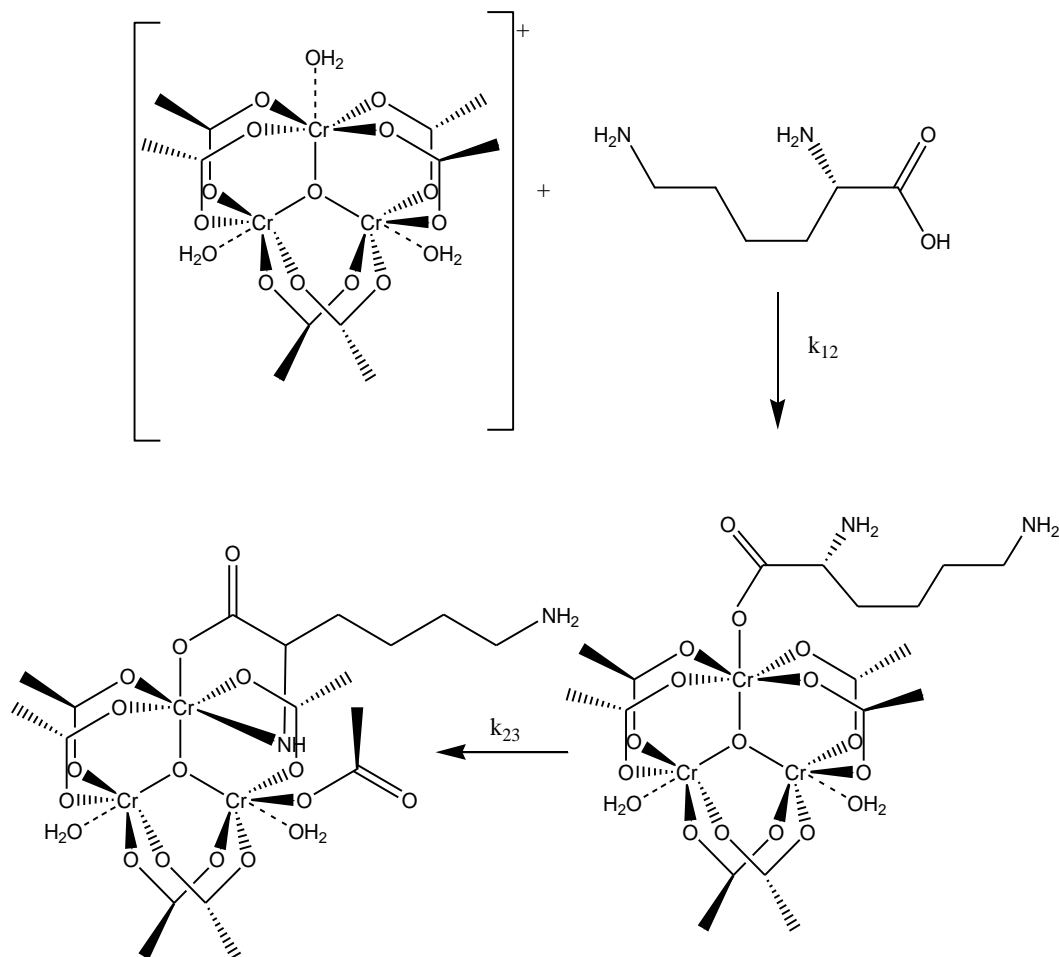
Una vez que hemos comprobado la aplicabilidad y buenos resultados del AH para la determinación de la Energía de Activación y el Factor Preexponencial en el mecanismo de reacción $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$ mediante el tratamiento de datos cinéticos no isotérmicos generados sintéticamente, hemos aplicado el AH para el tratamiento de datos experimentales. La reacción elegida para este propósito se basa en la ruptura del clúster de acetato de cromo trinuclear con una serie de aminoácidos monopróticos y dipróticos que actúan como ligando en medio acuoso⁽¹⁶¹⁾. En concreto hemos elegido la lisina teniendo en cuenta su gran importancia clínica:

- Beneficiosa para aquellas personas que presenten infecciones con Herpes.
- Existen conjugados de la lisina que resultan prometedoras para el tratamiento del cáncer, pues al parecer provocan que las células cancerosas se autodestruyan cuando el fármaco se utiliza en combinación con el uso de fototerapia, sin dañar a las células no cancerosas.
- La lisina, ingerida con L-carnitina, ayuda a que la carnitina disminuya notablemente los niveles de colesterol en sangre.
- La lisina estimula la liberación de la hormona del crecimiento. Esto ha hecho que se utilice, sola o combinada con otros aminoácidos, en niños para estimular el crecimiento y en ancianos para retrasar el envejecimiento.

La reacción ha sido estudiada experimentalmente para valores de pH comprendidos entre 3.5-5.5 y en un rango de temperatura de 45-60°C, siguiendo su evolución espectrofotométricamente. Considerando condiciones de reacción de pseudo-primer orden, las reacciones con estos ligandos pueden asimilarse a un mecanismo de reacción constituido por 2 etapas consecutivas, $B_1 \xrightarrow{k_{12}} B_2 \xrightarrow{k_{23}} B_3$, (Esquema VII.2.4) donde se supone que B_1 es un par iónico, B_2 es la especie intermedia y B_3 es el producto. Los valores de las constantes de velocidad de las dos etapas a diferentes temperaturas se determinaron a partir del ajuste de las curvas $[\ln(A_\infty - A_t)]$ vs. Tiempo. Los Parámetros Termodinámicos de Activación asociados a cada etapa de la reacción se calcularon utilizando la ecuación de Arrhenius y Eyring respectivamente considerando una variación hiperbólica de la temperatura ($m=1.57 \cdot 10^{-10} \text{ s}^{-1}$). El procedimiento se ha llevado a cabo considerando una variación de la temperatura comprendida entre 318.15 K y 333.15 K y una concentración inicial de reactivo de $3.00 \cdot 10^{-4} \text{ mol dm}^{-3}$. El intervalo de reacción estudiado ha sido aproximadamente dos veces el tiempo de vida media, lo que se corresponde a un tiempo de reacción de $9.00 \cdot 10^5$ segundos (10.4 días).

Los valores de las constantes de velocidad correspondientes a la primera y segunda etapa de la reacción son del orden: $k_{12} \approx 10^{-4} \text{ s}^{-1}$ y $k_{23} \approx 10^{-7} \text{ s}^{-1}$. Los valores de los Parámetros Termodinámicos de Activación determinados para las dos etapas de la reacción son los siguientes: $E_{12} = 70.7 \times 10^3 \text{ Jmol}^{-1}$, $A_{12} = 5.37 \times 10^7 \text{ s}^{-1}$, $E_{23} = 12.6 \times 10^4 \text{ Jmol}^{-1}$, $A_{23} = 1.81 \times 10^{14} \text{ s}^{-1}$,

$\Delta H_{12}^\ddagger = 68000 \text{ Jmol}^{-1}$, $\Delta S_{12}^\ddagger = -106 \text{ JK}^{-1}\text{mol}^{-1}$, $\Delta H_{23}^\ddagger = 123000 \text{ Jmol}^{-1}$ y $\Delta S_{23}^\ddagger = 19 \text{ JK}^{-1}\text{mol}^{-1}$. Los valores fuertemente negativos de ΔS_{12}^\ddagger y magnitud tan pequeña de ΔH_{12}^\ddagger correspondientes a la primera reacción indican un paso asociativo, mientras que los valores positivos de ΔS_{23}^\ddagger en el segundo paso muestran la disociación.



Esquema VII.2.4: Reacción entre el cluster $[\text{Cr}_3\text{O}(\text{OAc})_6(\text{H}_2\text{O})_3]^+$ con el ligando lisina.

La determinación de los Parámetros Termodinámicos de Activación correspondientes a la primera etapa de este mecanismo se realiza tal y como se ha indicado en apartados anteriores por lo que nos centraremos en la determinación de los PTA correspondientes a la segunda etapa del mecanismo E_{23} y A_{23} (sub-apartado VII.2.3.1.2) y ΔS_{23}^\ddagger y ΔH_{23}^\ddagger (sub-apartado VII.2.3.2.2).

VII.2.3.1.2.a-Predicción de E_{23} y A_{23} mediante ANN

i) Diseño de Experimentos

En este caso, en el Diseño de Experimentos (ED) están implicados dos factores E_{23} y A_{23} y las respuestas son los datos no isotérmicos cinéticos de la base de las curvas de Inputs (α_2/t). Para la realización el correspondiente Diseño de Experimentos hemos tomado valores

experimentales para E_{23} y A_{23} obtenidos a partir de datos bibliográficos ⁽¹⁶¹⁾: $E_{23} = 12.6 \times 10^4$ Jmol^{-1} y $A_{23} = 1.81 \times 10^{14} \text{ s}^{-1}$. La Figura VII.2.20 muestra el Diseño de Experimentos del tipo *Central Star Composite Experimental Design* (CSCED).

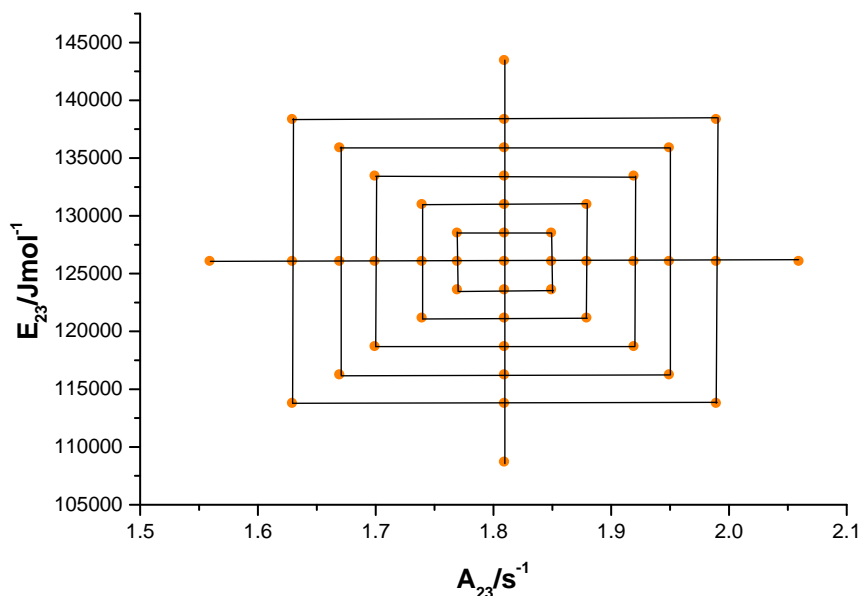


Figura VII.2.20: Diseño de Experimentos (CSCED) constituido por dos factores (A_{23}/E_{23}) y 45 puntos distribuidos en 9 niveles (puntos 1-37) y 4 sub-niveles (puntos 38-45) tomando como base de curvas la matriz de *Input* usada para el proceso de *Entrenamiento* de la Red.

ii) *Entrenamiento*

A partir de los pares de datos E_{23}/A_{23} correspondientes al ED, hemos generado 45 curvas cinéticas no isotérmicas (*Inputs*) considerando una variación de la temperatura hiperbólica. A continuación hemos llevado a cabo el proceso de *Entrenamiento* de la Red Neuronal Artificial (ANN) con el fin de encontrar la Red más adecuada que nos permita determinar E_{23} y A_{23} . Se realizó el *Entrenamiento* de la ANN variando sistemáticamente el número de nodos y la Configuración de las capas ocultas, considerando gran número de opciones posibles. La Red Neuronal idónea en este caso consta de 3 capas (Estructura) y una Configuración 10/10/14. En todos los casos se utilizó el mismo porcentaje de curvas para el *Entrenamiento*, *Validación* y *Test*, (80/10/10).

La representación conjunta de las regresiones *Outputs/Targets* de E_{23}/A_{23} se encuentran muy separados dentro de la gráfica, incluso aún más que en los casos anteriores ya considerados. Los órdenes de magnitud de ambos parámetros son muy diferentes, $E_{23} \approx 10^5$ y $A_{23} \approx 10^{14}$. En el caso de las desviaciones de A_{23} existe, aparentemente, una pésima linealidad y en cuanto a las desviaciones de E_{23} todos los valores de dispersión se superponen y producen un único punto. Como ya hemos explicado, esto es debido a las diferentes magnitudes de las escalas de ambos parámetros (Figura VII.2.21).

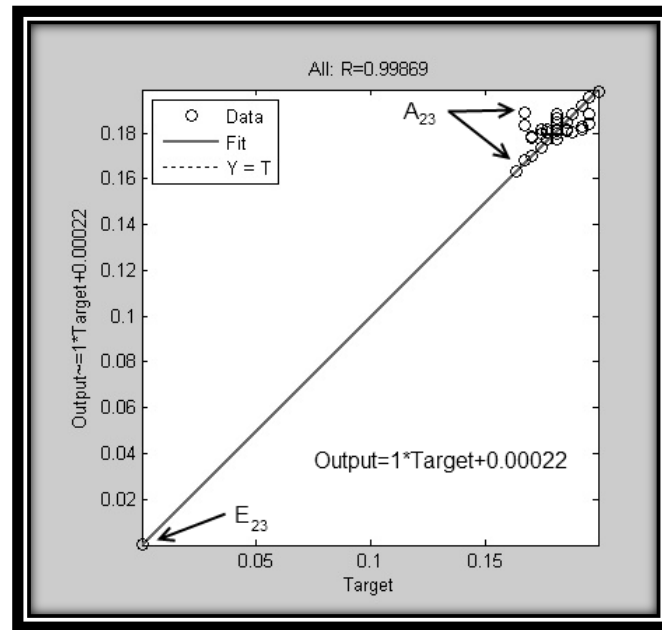


Figura VII.2.21: Regresión lineal de las matrices de *Outputs vs Targets*, cuando ambos parámetros (E_{23}/A_{23}) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* ($9.00.10^5$, 3, 2).

Aparentemente estos parámetros se ajustan a una recta pero si tenemos en cuenta las regresiones *Outputs/Targets* de E_{23} y A_{23} por separado se observa una gran dispersión en ambos parámetros (Figuras VII.2.22.a y b), pero en mayor medida en A_{23} , que hace que los resultados sean inaceptables mediante la aplicación únicamente de ANN, lo que implica que sea necesario utilizar otro algoritmo, en nuestro caso AGDC para completar el AH.

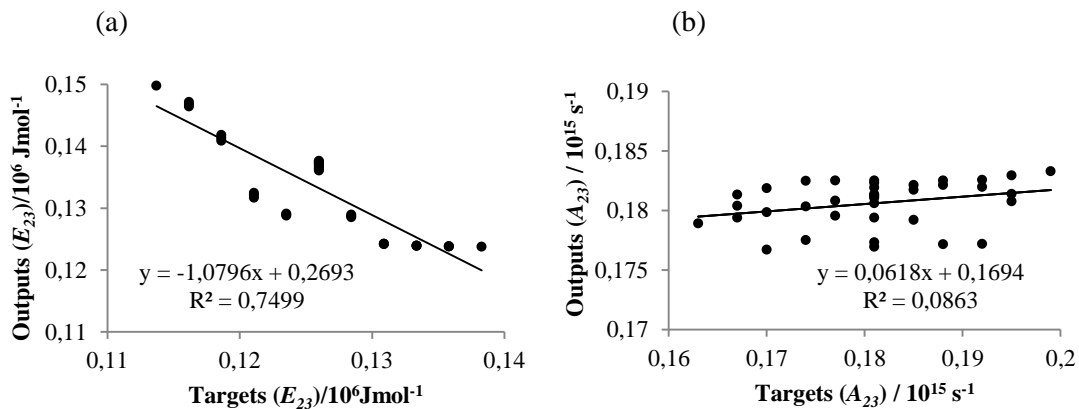


Figura VII.2.22.a y b: Regresión lineal de las matrices de *Outputs vs Targets*, exclusivamente en el caso de E_{23} (a) y A_{23} (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 3 *hidden layers* ($9.00.10^5$, 3, 2).

iii) *Predicción*

El último paso de la metodología ANN consiste en realizar el proceso de *Predicción* de E_{23} y A_{23} a partir de datos cinéticos experimentales procedentes de experiencias cinéticas en las que su valor es desconocido. El proceso de *Entrenamiento* de la ANN se ha realizado considerando un gran número de Configuraciones posibles, mostrándose en la Tabla VII.2.12 se muestran los resultados obtenidos para las Configuraciones más significativas correspondientes al punto central de las 45 curvas de ED. El par de valores de *Outputs*, E_{23}/A_{23} y las correspondientes desviaciones (“*Des. %*”) indican que es necesario mejorar los resultados mediante la aplicación del algoritmo AGDC (segunda etapa del AH).

VII.2.3.1.2.b- Optimización de A_{23} y E_{23} mediante AGDC

Puesto que es necesario mejorar los valores de E_{23} y A_{23} determinados con ANN, hemos aplicado el algoritmo de Optimización Matemática AGDC para obtener un valor óptimo de los mismos. El proceso de optimización se ha llevado a cabo mediante la aplicación del programa **KINNOISOT(AGDC)**, de igual manera que en los casos anteriores, utilizando como estimaciones iniciales los valores obtenidos con ANN. La segunda parte de la Tabla VII.2.12 muestra los valores de los parámetros optimizados y las desviaciones de los valores reales (“*Des. %*”). Comparando los valores de las desviaciones (columnas octava y novena) de ambos parámetros determinados mediante el AH con los calculados por la aplicación ANN (columnas cuarta y quinta) se confirma la mejora en los resultados. Teniendo en cuenta los valores obtenidos para las desviaciones, la Arquitectura que consigue los mejores resultados para llevar a cabo el proceso de *Predicción* es aquella con 3 capas y una Configuración 10/10/14 nodos, respectivamente. En este caso concreto, podemos observar que al aplicar el AGDC la desviación de la Energía de Activación no se modifica permaneciendo constante en todos los casos, de tal forma que hemos tratado de elegir aquella Arquitectura en la que el cálculo de los errores fuera el menor tanto para ANN como para AGDC.

Config.	ANN				(ANN - AGDC)			AH
	$A_{23} / 10^{14} \text{ s}^{-1}$	$E_{23} / 10^5 \text{ Jmol}^{-1}$	Des. % (A_{23})	Des. % (E_{23})	$A_{23} / 10^{14} \text{ s}^{-1}$	$E_{23} / 10^5 \text{ Jmol}^{-1}$	Des. % (A_{23})	Des. % (E_{23})
7/10	1.8151	1.1930	$4.9871 \cdot 10^{-1}$	5.3163	1.8131	1.1930	$-2.7951 \cdot 10^{-1}$	5.3162
10/8	1.8142	1.3272	$6.0382 \cdot 10^{-1}$	-5.3406	1.8142	1.3273	$-2.3482 \cdot 10^{-1}$	-5.3401
10/10	1.8119	1.4265	$7.4679 \cdot 10^{-1}$	-13.217	1.8128	1.4265	$-1.0248 \cdot 10^{-1}$	-13.217
10/11	1.8137	1.1913	-1.0254	5.4514	1.8147	1.1913	$-2.0487 \cdot 10^{-1}$	5.4514
10/12	1.8113	1.1657	$4.9593 \cdot 10^{-1}$	7.4846	1.8113	1.1657	$-7.0313 \cdot 10^{-2}$	7.4846
10/14	1.8088	1.3846	$6.1918 \cdot 10^{-1}$	-9.8910	1.8098	1.3846	$6.9498 \cdot 10^{-2}$	-9.8910
10/20	1.8115	1.2861	$-5.7425 \cdot 10^{-1}$	-2.0719	1.8125	1.2861	$-8.4025 \cdot 10^{-2}$	-2.0719
13/10	1.7950	1.1106	-1.3458	11.853	1.7950	1.1106	$8.2860 \cdot 10^{-1}$	11.853
14/10	1.8085	1.1178	$4.8265 \cdot 10^{-1}$	11.289	1.8085	1.1178	$8.3585 \cdot 10^{-2}$	11.289
20/10	1.7933	1.3579	$-9.8133 \cdot 10^{-1}$	-7.7730	1.7933	1.3579	$9.2453 \cdot 10^{-1}$	-7.7730
10/10/14	1.8138	1.2714	$-2.1188 \cdot 10^{-1}$	$-9.0834 \cdot 10^{-1}$	1.8148	1.2714	$-2.1188 \cdot 10^{-1}$	$-9.0834 \cdot 10^{-1}$

Tabla VII.2.12: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando A_{23} y E_{23} son computados.

Por último, se aplicó el Algoritmo Híbrido para determinar los valores de A_{23} y E_{23} correspondientes a nuevas experiencias cinéticas no isotérmicas. En primer lugar se llevó a cabo el proceso de *Predicción* con la ANN óptima y los valores obtenidos son sometidos a un proceso posterior de optimización mediante la aplicación del algoritmo AGDC, obteniendo así los valores optimizados de A_{23}/E_{23} . La Tabla VII.2.13 contiene los resultados obtenidos tras la aplicación del AH completo que nos proporciona los valores desconocidos de los parámetros.

(ANN + AGDC)		AH			
Valores Reales (A_{23})/ 10^{14} s^{-1}	Valores Reales (E_{23})/ 10^5 Jmol^{-1}	Valores Optimizados (A_{23})/ 10^{14} min^{-1}	Valores Optimizados (E_{23})/ 10^5 Jmol^{-1}	% Des. (A_{23})	% Des. (E_{23})
1.8199	1.5299	1.8199	1.5299	5.2112	-28.964
1.8173	1.3696	1.8173	1.3696	-6.5722	-2.6932
1.8033	1.4829	1.8033	1.4829	4.0785	-22.472
1.8040	1.3524	1.8040	1.3524	-3.6787	-3.3073
1.8142	1.4870	1.8142	1.4870	1.9364	-20.369
1.8049	1.3160	1.8049	1.3160	-1.9707	-2.4489
1.8040	1.3509	1.8040	1.3509	$3.3280 \cdot 10^{-1}$	-3.1919
1.8138	1.2714	1.8138	1.2714	$-2.1188 \cdot 10^{-1}$	$-9.0834 \cdot 10^{-1}$

Tabla VII.2.13: Resultados de los 8 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (9.00.10⁵, 3, 2) con 3 *hidden layers* y Configuración 10/10/14, cuando A_{23}/E_{23} son computados después de la aplicación de AH (ANN-AGDC).

VII.2.3.2.-Determinación de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger

VII.2.3.2.1.-Tratamiento de datos sintéticos

VII.2.3.2.1.a-Predicción de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger mediante ANN

i) Diseño de Experimentos

El Diseño de Experimentos realizado es del tipo CSED (Figura VII.2.23), en este caso los *factores* están formados por el conjunto de parámetros $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ y las *respuestas* en las cinéticas no isotérmicas son los datos de la base de curvas de *Input* (α_2/t , y α_3/t). Los valores centrales de ambos factores son los siguientes: $\Delta H_{23}^\ddagger = 45172 \text{ Jmol}^{-1}$ y $\Delta S_{23}^\ddagger = -163.1 \text{ JK}^{-1}\text{mol}^{-1}$.

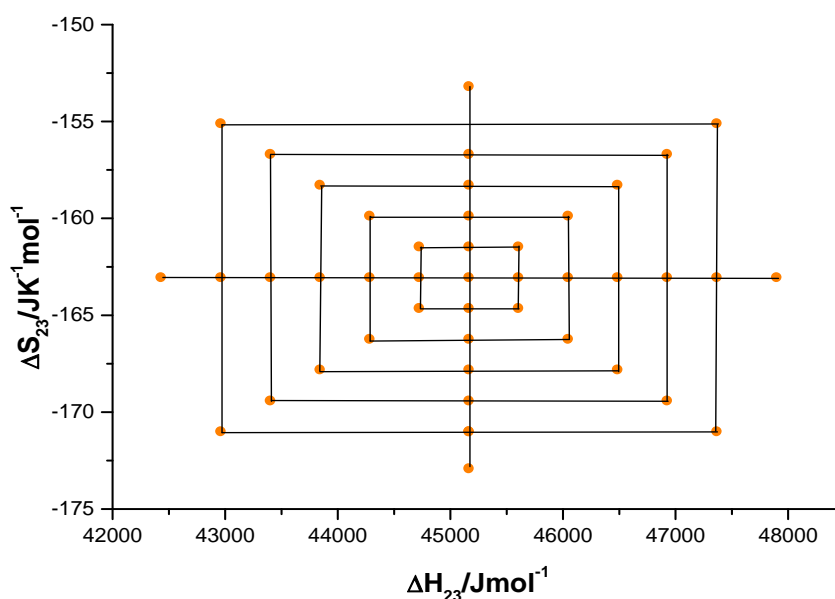


Figura VII.2.23: Diseño de Experimentos (CSED) constituido por dos *factores* ($\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$) y 45 puntos distribuidos en 9 niveles (puntos 1-37) y 4 sub-niveles (puntos 38-45) tomando como base de curvas la matriz de *Input* usada para el proceso de *Entrenamiento* de la Red.

ii) Entrenamiento

A partir del Diseño de Experimentos que hemos realizado para este caso, se generan las matrices de *Inputs* formadas por 45 curvas cinéticas no isotérmicas con las siguientes características: constan de 100 pares de datos cinéticos (α_2/t y α_3/t), el valor del *noise* es del orden del error experimental ($\pm 1.0 \cdot 10^{-4}$), el intervalo de tiempo hasta que la reacción alcance el 75% y la función T/t de tipo polinómica. La Arquitectura óptima de la Red Neuronal se determinó a partir del proceso de *Entrenamiento* con porcentajes de *Training/Validation/Testing* 80/10/10. La Arquitectura óptima de la Red para realizar la *Predicción* es aquella con un valor menor de MSE ($\approx 10^{-6}$) por lo que comprobamos que la Red Neuronal más apta consta de 2 capas (Estructura) y dentro de esa capa los nodos óptimos son 14/10 (Configuración).

La Figura VII.2.24 muestra la representación gráfica de los resultados de los procesos de *Training*, *Validation* y *Testing* (“ALL”) para una Red con una Estructura óptima (100, 2, 2)

y Configuración 14/10. En la gráfica se representan simultáneamente ambos parámetros (ΔH_{23}^{\neq} y ΔS_{23}^{\neq}) y se lleva a cabo la regresión lineal: $Output=1.00xTarget+1.30.10^{-6}$, obteniéndose un valor de $R=0.99999$ que corresponde a una buena linealidad entre los elementos de las matrices $Output/Target$. Estos resultados podrían ser inicialmente aceptables, sin embargo se deben tener en cuenta unas consideraciones importantes: el orden de magnitud de los valores de los parámetros es muy diferente ($\Delta H_{23}^{\neq} \approx 10^4$ y $\Delta S_{23}^{\neq} \approx 10^2$), y lógicamente al estar representados en la misma escala, los puntos que aparecen en el gráfico forman dos grupos separados por una gran distancia entre ellos. Además todos los valores correspondientes a ΔS_{23}^{\neq} se superponen en el mismo punto. Esta situación enmascara los efectos reales de la dispersión, lo que provoca que la recta de regresión cuando ambos PTA están representados conjuntamente sea correcta.

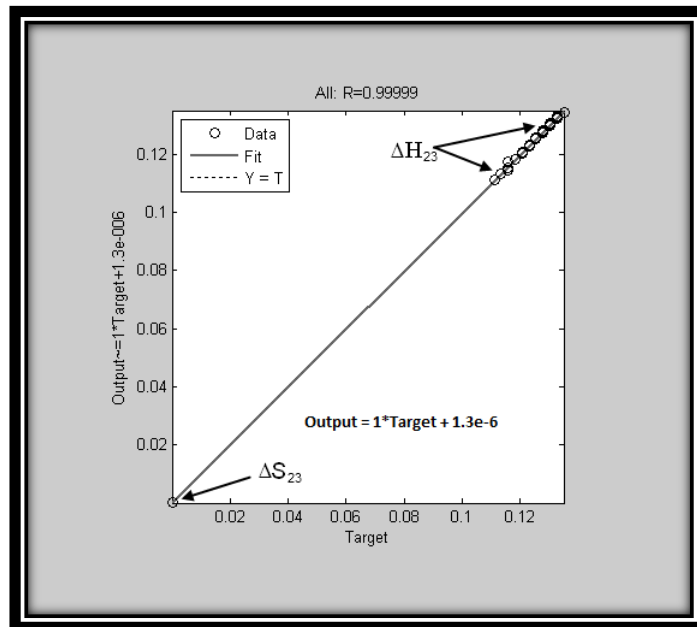


Figura VII.2.24: Regresión lineal de las matrices de *Outputs vs Targets*, cuando ambos parámetros ($\Delta H_{23}^{\neq}/\Delta S_{23}^{\neq}$) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 2 *hidden layers* (100, 2, 2).

Si tenemos en cuenta las rectas de regresión *Outputs/Targets* de cada parámetro por separado utilizando la escala apropiada, existe una gran dispersión para el caso de ΔS_{23}^{\neq} (Figura VII.2.25.b), correspondiente a los puntos superpuestos que se observan en la Figura VII.2.24. Los valores obtenidos son: 0.1285 para R^2 , 0.3314 para la pendiente (muy lejos del valor = 1) y 0.0383 para la ordenada en el origen (muy distinta de 0). Estos valores manifiestan claramente una correlación muy pobre entre *Outputs/Targets*, que hace inaceptable los valores de ΔS_{23}^{\neq} determinados por ANN. Sin embargo, los resultados obtenidos para ΔH_{23}^{\neq} , demuestran tener una linealidad aceptable como mostramos en la Figura VII.2.25.a.

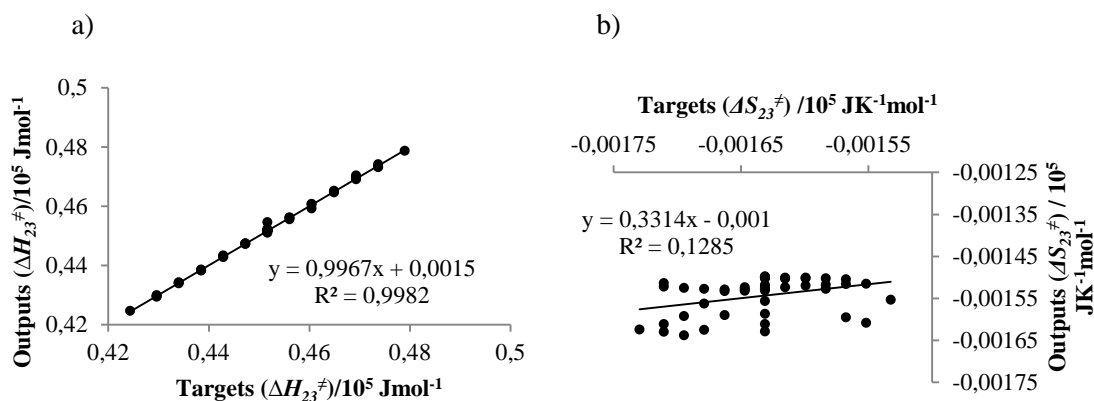


Figura VII.2.25.a y b: Regresión lineal de las matrices de *Outputs* vs *Targets*, exclusivamente en el caso de ΔH_{23}^{\ddagger} (a) y ΔS_{23}^{\ddagger} (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 2 *hidden layers* (100, 2, 2).

Los resultados obtenidos indican la necesidad de aplicar el AH en su totalidad, es decir, los valores obtenidos para los PTA a partir de la metodología ANN, serán utilizados como estimaciones iniciales para el algoritmo AGDC, que nos proporcionará valores mejorados para ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} .

iii) *Predicción*

Una vez determinada Arquitectura óptima de la Red Neuronal de para la *Predicción* de ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} , se utiliza ésta ANN para el tratamiento de nuevas curvas cinéticas, diferentes de las consideradas en el proceso de *Entrenamiento* y para las que desconocemos los valores de ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} . El proceso de *Entrenamiento* se llevó a cabo para un gran conjunto de Redes Neuronales, considerando un gran número de posibles Configuraciones para las Arquitecturas con 2 y 3 capas Ocultas. En la Tabla VII.2.14 se muestran los resultados más relevantes, obtenidos para las Configuraciones más significativas correspondientes al punto central de las 45 curvas de ED. En la primera parte se muestran los valores de *Outputs*, $\Delta H_{23}^{\ddagger}/\Delta S_{23}^{\ddagger}$ y las correspondientes desviaciones (“*Des. %*”). Los valores obtenidos para los errores de los parámetros indican que es necesario mejorar los resultados mediante la aplicación de la segunda etapa del Algoritmo Híbrido (AH) que constituye la aplicación de AGDC.

VII.2.3.1.1.b- Optimización de ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} mediante AGDC

En el proceso de *Predicción* realizado con ANN se obtienen los valores de ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} que posteriormente son sometidos a un proceso de mejora y refinamiento mediante la aplicación del algoritmo AGDC. La Tabla VII.2.14 muestra los valores de ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} obtenidos mediante la aplicación de únicamente ANN y los que se obtiene tras la aplicación del Algoritmo Híbrido (ANN-AGDC). Se recogen los resultados correspondientes a las configuraciones más significativas correspondientes al punto central de las 45 que componen el ED. En la primera parte de la Tabla se muestran los valores de ΔH_{23}^{\ddagger} y ΔS_{23}^{\ddagger} obtenidos con

ANN y las desviaciones en % (“Des. %”) de estos valores con respecto a los que sirvieron para generar los datos. Los valores de ΔH_{23}^\ddagger y ΔS_{23}^\ddagger optimizados con AGDC y las desviaciones de los valores reales (“Des. %”) se encuentran en la 2ª parte de la tabla. Después de la aplicación del AH observamos una gran mejora cuando se comparan los de valores de las desviaciones (columnas octava y novena) de ambos parámetros con los calculados por la aplicación ANN (columnas cuarta y quinta).

Config.	ANN				(ANN + AGDC) AH			
	$\Delta H_{23}^\ddagger/10^4$ Jmol ⁻¹	$\Delta S_{23}^\ddagger/$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{23}^\ddagger)	Des. % (ΔS_{23}^\ddagger)	$\Delta H_{23}^\ddagger/10^4$ Jmol ⁻¹	$\Delta S_{23}^\ddagger/$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{23}^\ddagger)	Des. % (ΔS_{23}^\ddagger)
8/10	4.5016	-163.72	3.4516.10 ⁻¹	-3.7829.10 ⁻¹	4.5000	-163.65	3.8110.10 ⁻¹	-3.3911.10 ⁻¹
10/9	4.4914	-164.02	5.7044.10 ⁻¹	-5.6553.10 ⁻¹	4.4902	-163.97	5.9722.10 ⁻¹	-5.3344.10 ⁻¹
10/11	4.4979	-160.56	4.2519.10 ⁻¹	1.5559	4.5449	-162.20	-6.1429.10 ⁻¹	0.54984.10 ⁻¹
10/12	4.4927	-179.05	5.4107.10 ⁻¹	-9.7783	4.4685	-164.67	1.0782	-9.6032.10 ⁻¹
10/15	4.5143	-136.32	6.2543.10 ⁻²	16.419	4.6457	-158.95	-2.8456	2.5455
10/20	4.5212	-178.30	-8.9812.10 ⁻²	-9.3250	4.5329	-162.60	-3.4829.10 ⁻¹	3.0936.10 ⁻¹
13/10	4.5002	-160.48	3.7522.10 ⁻¹	1.6075	4.5774	-162.13	-6.6794.10 ⁻¹	5.9755.10 ⁻¹
14/10	4.5430	-163.73	-5.7140.10⁻¹	-3.8617.10⁻¹	4.5196	-163.02	-5.3196.10⁻²	4.8115.10⁻²
15/10	4.5239	-161.74	-1.4999.10 ⁻¹	8.3355.10 ⁻¹	4.5416	-162.32	-5.4056.10 ⁻¹	4.8130.10 ⁻¹
9/10/10	4.5061	-161.99	0.43631.10 ⁻¹	6.7740.10 ⁻¹	4.5285	-162.74	-2.5085.10 ⁻¹	2.2371.10 ⁻¹
10/9/10	4.5271	-181.53	0.57041.10 ⁻¹	-11.299	4.6161	-159.91	-2.1899	1.9530
10/10/10	4.5210	-162.62	0.15820.10 ⁻¹	2.9162.10 ⁻¹	4.5268	-162.79	-2.1258.10 ⁻¹	-1.8898.10 ⁻¹
10/10/13	4.4894	-161.823	-8.9894.10 ⁻²	7.8101.10 ⁻¹	4.5222	-162.94	-1.1092.10 ⁻¹	1.0026.10 ⁻¹
10/10/14	4.5145	-160.37	-7.6705.10 ⁻¹	1.6744	4.5567	-161.83	-8.7547.10 ⁻¹	7.8153.10 ⁻¹
10/12/10	4.5064	-163.62	-1.4994.10 ⁻¹	-3.2089.10 ⁻¹	4.5038	-163.53	2.9738.10 ⁻¹	-2.6443.10 ⁻¹
10/15/10	4.5121	-165.07	1.1131.10 ⁻¹	-1.2081	4.4830	-164.19	7.5660.10 ⁻¹	-6.7092.10 ⁻¹
10/20/10	4.5263	-182.96	-2.0263.10 ⁻¹	-12.176	4.5745	-161.27	-1.2682	1.1213

Tabla VII.2.14: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando ΔH_{23}^\ddagger y ΔS_{23}^\ddagger son computados.

La mejora en los valores de los Parámetros Termodinámicos de Activación se aprecia claramente cuando se realiza una representación conjunta de los residuales de α_2 , calculadas con los valores de los parámetros $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$, que se obtienen a partir de la aplicación de ANN [VII.2.1] y los valores determinados por la aplicación del algoritmo Híbrido [VII.2.2]. En la Figura VII.2.26 podemos apreciar esta mejora, ya que el valor máximo se alcanza cuando ha transcurrido el 40% de la reacción mediante la comparación de los valores de Res (α)_{AH}=0.0013 y Res (α)_{ANN}=0.388.

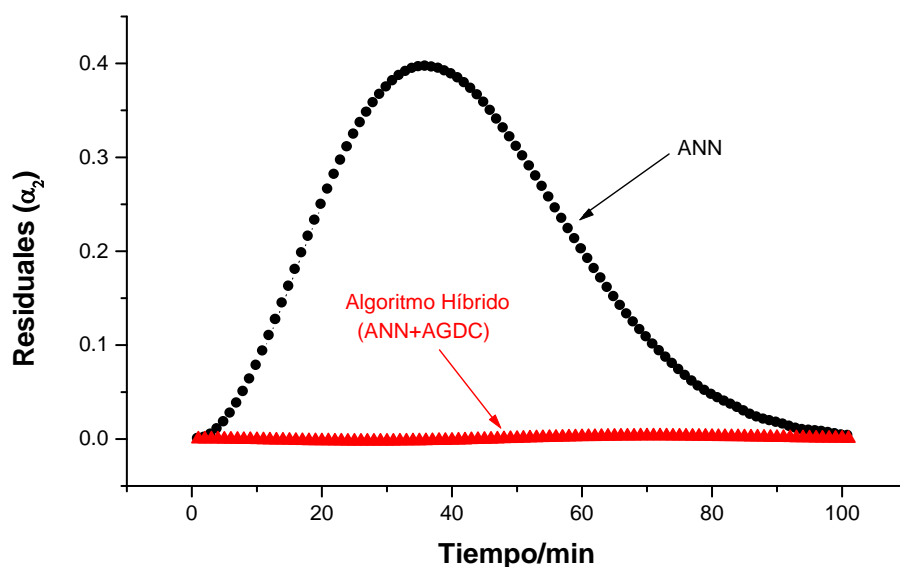


Figura VII.2.26: Figura de las residuales de α_2 calculados a partir de la ecuación [VII.2.1] con los valores de los parámetros ΔH_{23}^\ddagger y ΔS_{23}^\ddagger obtenidos mediante la aplicación de la ANN y los obtenidos a partir de la ecuación [VII.2.2] después de la aplicación del algoritmo completo.

Por último hemos aplicado el Algoritmo Híbrido para determinar los valores $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ correspondientes a nuevas experiencias cinéticas no isotérmicas, generadas a partir de pares de valores comprendidos dentro de los intervalos del ED. El proceso de *Predicción* se llevó a cabo con la Red Neuronal óptima para 8 curvas cinéticas no isotérmicas. Los resultados obtenidos para la *Predicción* de los 8 pares de valores $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ mediante ANN, son sometidos a un proceso posterior de optimización mediante la aplicación del algoritmo AGDC, teniendo en cuenta que los porcentajes de desviaciones (“% Des.”) poseen una magnitud similar al de las desviaciones del proceso de *Entrenamiento* de la Red Neuronal óptima, podemos aceptar los valores optimizados de ΔH_{23}^\ddagger y ΔS_{23}^\ddagger como definitivos para los 4 procesos de *Predicción*. Los resultados obtenidos se muestran en la Tabla VII.2.15.

(ANN		-	AGDC)	AH	
Valores Reales (ΔH_{23}^\ddagger)/ 10^4 Jmol ⁻¹	Valores Reales (ΔS_{23}^\ddagger)/JK ⁻¹ mol ⁻¹	Valores Optimizados (ΔH_{23}^\ddagger)/ 10^4 Jmol ⁻¹	Valores Optimizados (ΔS_{23}^\ddagger)/JK ⁻¹ mol ⁻¹	% Des. (ΔH_{23}^\ddagger)	% Des. (ΔS_{23}^\ddagger)
4.3408	-159.56	4.2976	-158.17	9.9996.10 ⁻¹	-9.1361.10 ⁻¹
4.7031	-163.81	4.7811	-166.68	-1.8692	1.6402
4.3832	-152.75	4.4701	-155.55	-1.9387	1.7222
4.4330	-150.11	4.5482	-156.05	-2.6887	2.4209
4.6072	-159.01	4.6994	-163.33	-2.0446	1.7745
4.4734	-150.12	4.5650	-158.55	-2.0531	1.8465
4.5551	-153.21	4.7207	-159.59	-3.4960	3.0962
4.5172	-152.65	4.6775	-157.94	-3.5481	3.1664

Tabla VII.2.15: Resultados de los 8 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (100, 2, 2) con 2 *hidden layers* y Configuración 14/10, cuando $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ son computados después de la aplicación de AH (ANN-AGDC).

VII.2.3.2.2.-Tratamiento de datos experimentales

VII.2.3.2.2.a-Predicción de ΔS_{23}^\ddagger y ΔH_{23}^\ddagger mediante ANN

i) Diseño de Experimentos

Al igual que en los demás casos, el Diseño de Experimentos realizado es del tipo *Central Star Composite Experimental Design* (CSCED), los *factores* están formados por el conjunto de parámetros $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ y las *respuestas* son los datos de la base de curvas de *Input* (α_2/t y α_3/t). Los valores del punto central, adquiridos en bibliografía ⁽¹⁶¹⁾, son los siguientes: $\Delta H_{23}^\ddagger = 123000 \text{ Jmol}^{-1}$ y $\Delta S_{23}^\ddagger = 19 \text{ JK}^{-1}\text{mol}^{-1}$. Además es necesario tener en cuenta que los valores de ΔH^\ddagger y ΔS^\ddagger para la primera etapa de la reacción son: $\Delta H_{12}^\ddagger = 6800 \text{ Jmol}^{-1}$ y $\Delta S_{12}^\ddagger = -106 \text{ JK}^{-1}\text{mol}^{-1}$. La Figura VII.2.27 muestra el Diseño de Experimentos realizado para este caso a partir del cual se han generado las 37 curvas de *Inputs* para realizar el proceso de *Entrenamiento* de la Red.

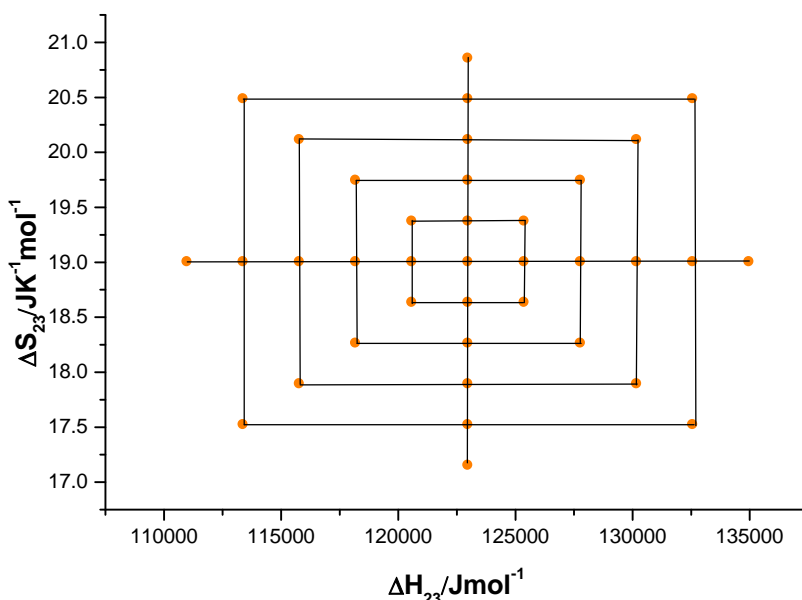


Figura VII.2.27: Diseño de Experimentos (CSCED) constituido por dos *factores* ($\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$) y 37 puntos distribuidos en 9 niveles y 3 sub-niveles tomando como base de curvas la matriz de *Input* usada para el proceso de *Entrenamiento* de la Red.

ii) Entrenamiento

Una vez que hemos generado las 37 curvas experimentales (*Inputs*) considerando una variación de la temperatura polinómica, llevamos a cabo el proceso de *Entrenamiento* de ANN con el fin de encontrar la Red más adecuada que nos permita determinar ΔH_{23}^\ddagger y ΔS_{23}^\ddagger . Para determinar la Arquitectura óptima de la Red Neuronal se realizó el *Entrenamiento* variando el número de nodos y la Configuración de las capas ocultas sistemáticamente, considerando todas las opciones posibles. Así, tras la experimentación comprobamos que la Red Neuronal más apta consta de 1 capa (Estructura) y dentro de esa capa el número de nodos óptimo es 6

(Configuración). En todos los casos se ha utilizado el mismo porcentaje de todas las curvas dentro de las tratadas para el *Entrenamiento, Validación y Test*, (80/10/10). Este es el único caso en el que la Arquitectura de la Red es más sencilla, ya que al realizar el proceso de *Entrenamiento* con Estructuras más complejas y un número más elevado de nodos, no se modifica significativamente el valor del MSE ($\approx 10^{-8}$).

Cuando se representan de manera conjunta las regresiones *Outputs /Targets* de ΔH_{23}^{\neq} y ΔS_{23}^{\neq} (Figura VII.2.28), al presentar órdenes de magnitud muy diferentes ($\Delta H_{23}^{\neq} \approx 10^5$ y $\Delta S_{23}^{\neq} \approx 10^1$) por lo que se encuentran muy separados dentro de la gráfica. En el caso de las desviaciones de ΔH_{23}^{\neq} existe, aparentemente, una buena linealidad, lo cual corroboramos al realizar la regresión individual (Figura VII.2.29.a). En el caso de las desviaciones ΔS_{23}^{\neq} la Figura VII.2.29.b muestra la superposición de todos los valores de dispersión dando lugar a un único punto como consecuencia de las diferentes magnitudes de las escalas de ambos parámetros. Una vez más, quedan enmascarados los verdaderos efectos de la dispersión haciendo que los parámetros se ajusten bien a una recta de regresión que podría llevarnos a considerar aceptable esta gráfica. Como ya hemos comentado anteriormente, ésto no es así y hay que considerar los parámetros de forma individual.

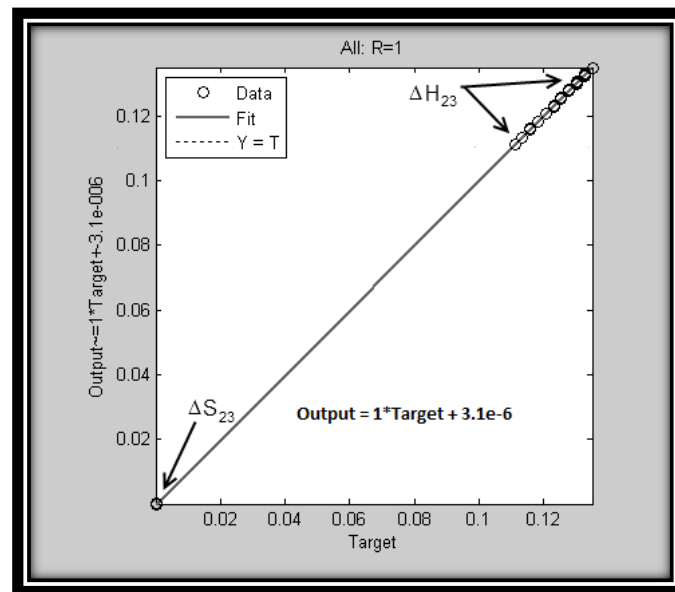


Figura VII.2.28: Regresión lineal de las matrices de *Output vs Target*, cuando ambos parámetros ($\Delta S_{23}^{\neq}/\Delta H_{23}^{\neq}$) son representados en conjunto durante el proceso de *Entrenamiento* de la ANN con una Arquitectura de 1 *hidden layer* ($9.00.10^5$, 1, 2).

Representando por separado las regresiones *Outputs/Targets* de ΔH_{23}^{\neq} (Figura VII.2.29.a) y ΔS_{23}^{\neq} (Figura VII.2.29.b) se observa una gran dispersión en el caso de ΔS_{23}^{\neq} (Figura VII.2.29.b) que hace que los resultados sean inaceptables mediante la aplicación únicamente de ANN, por lo que es necesario utilizar otro algoritmo.

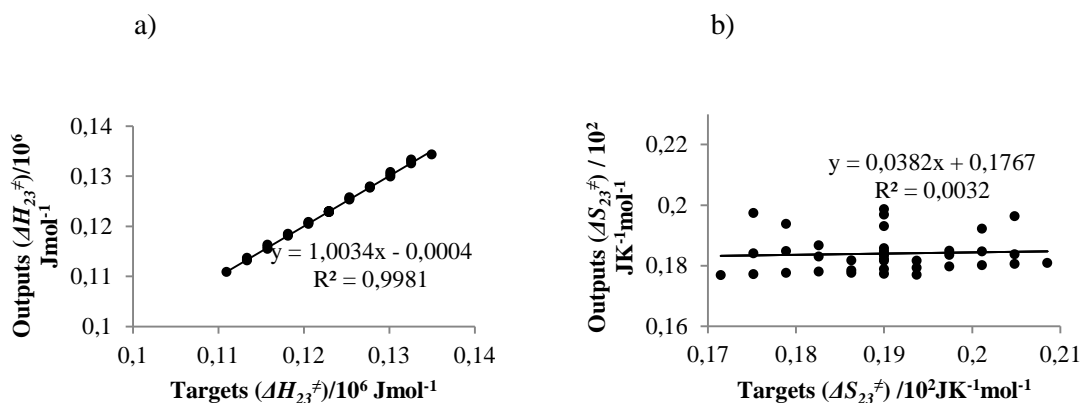


Figura VII.2.29.a y b: Regresión lineal de las matrices de *Output vs Target*, exclusivamente en el caso de ΔH_{23}^\ddagger (a) y ΔS_{23}^\ddagger (b) para el proceso de *Entrenamiento* de la ANN con una Arquitectura de 1 *hidden layer* ($9.00 \cdot 10^5$, 1, 2).

En este caso también hemos realizado el proceso de *Entrenamiento* utilizando como *Inputs* las curvas cinéticas no isotérmicas (α_3/t) correspondientes al producto de reacción B₃, con objeto de comprobar si se observa alguna variación en los resultados obtenidos y si pueden considerarse igualmente válidos. Después de realizar un procedimiento análogo se llega a obtener conclusiones idénticas que cuando se lleva a cabo el proceso de *Entrenamiento* utilizando como *Inputs* las curvas cinéticas no isotérmicas (α_2/t), con la única diferencia relativa a la Configuración óptima ya que en este caso resulta ser 5 nodos.

iii) Predicción

El proceso de *Predicción* de ΔH_{23}^\ddagger y ΔS_{23}^\ddagger a partir de datos cinéticos experimentales procedentes de experiencias cinéticas en las que su valor es desconocido se lleva a cabo con la ANN óptima determinada en el proceso de *Entrenamiento*. La ANN ha sido *entrenada* considerando un gran número de Configuraciones posibles para las Arquitecturas con 1 capa Oculta, descartando aquellas con dos o más capas ya que no mejoraban el proceso de *Aprendizaje*. En la Tabla VII.2.16 se muestran los resultados obtenidos para las Configuraciones más significativas correspondientes al punto central de las 37 curvas de ED. El par de valores de *Outputs* $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ y las correspondientes desviaciones ("*Des. %*") indican que es necesario mejorar los resultados mediante la aplicación del algoritmo AGDC.

VII.2.3.2.2.b- Optimización de ΔH_{23}^\ddagger y ΔS_{23}^\ddagger mediante AGDC

La segunda parte de la Tabla VII.2.16 muestra los valores de $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ optimizados después de aplicar AGDC y las desviaciones de los valores reales ("*Des. %*").

Config.	ANN				(ANN - AGDC) AH			
	$\Delta H_{23}^{\ddagger}/10^5$ Jmol ⁻¹	$\Delta S_{23}^{\ddagger}/$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{23}^{\ddagger})	Des. % (ΔS_{23}^{\ddagger})	$\Delta H_{23}^{\ddagger}/10^5$ Jmol ⁻¹	$\Delta S_{23}^{\ddagger}/$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{23}^{\ddagger})	Des. % (ΔS_{23}^{\ddagger})
5	1.2307	16.956	-5.2924.10 ⁻²	10.760	1.2270	18.085	2.4460.10 ⁻¹	4.8174
6	1.2304	19.781	-3.0505.10⁻²	-4.1098	1.2347	19.449	-1.1937.10⁻¹	-2.3625
7	1.2298	17.248	1.6711.10 ⁻²	9.2234	1.2270	18.096	2.4090.10 ⁻¹	4.7587
8	1.2298	20.862	1.6479.10 ⁻²	-9.7972	1.2330	19.905	-2.4050.10 ⁻¹	-4.7615
9	1.2301	15.864	-4.0925.10 ⁻³	16.506	1.2250	17.452	4.1270.10 ⁻¹	8.1394
10	1.2299	15.491	4.9065.10 ⁻³	18.471	1.2247	17.253	4.6637.10 ⁻¹	9.1970
12	1.2292	17.886	6.7485.10 ⁻²	5.8637	1.2278	18.316	1.8178.10 ⁻¹	3.5994
15	1.2301	14.645	-7.8094.10 ⁻³	22.921	1.2298	16.860	5.7118.10 ⁻¹	11.265
20	1.2306	13.510	-4.6183.10 ⁻²	28.895	1.2215	16.387	6.9365.10 ⁻¹	13.755
22	1.2303	16.025	-2.1074.10 ⁻²	15.659	1.2253	17.565	3.8303.10 ⁻¹	7.5515

Tabla VII.2.16: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura 1 *hidden layer* y distintas Configuraciones, cuando ΔS_{23}^{\ddagger} y ΔH_{23}^{\ddagger} son computados.

En la Tabla VII.2.16 se puede observar que al aplicar el algoritmo AGDC la desviación de ΔS_{23}^{\ddagger} disminuye considerablemente, pero la desviación de ΔH_{23}^{\ddagger} aumenta con respecto a los resultados obtenidos con ANN. En vista de estos resultados hemos aplicado el algoritmo AGDC única y exclusivamente para la mejora de los valores de ΔS_{23}^{\ddagger} obtenidos mediante ANN. La Tabla VII.2.17 muestra los resultados obtenidos para este caso.

Config.	ANN				ANN - AGDC	
	$\Delta H_{23}^{\ddagger}/10^5$ Jmol ⁻¹	$\Delta S_{23}^{\ddagger}/$ JK ⁻¹ mol ⁻¹	Des. % (ΔH_{23}^{\ddagger})	Des. % (ΔS_{23}^{\ddagger})	$\Delta S_{23}^{\ddagger}/$ JK ⁻¹ mol ⁻¹	Des. % (ΔS_{23}^{\ddagger})
5	1.2307	16.956	-5.2924.10 ⁻²	10.7601	19.383	-9.6541.10 ⁻¹
6	1.2304	19.781	-3.0505.10 ⁻²	-4.1098	19.099	-5.1888.10 ⁻¹
7	1.2298	17.248	1.6711.10 ⁻²	9.2234	18.920	4.2204.10 ⁻¹
8	1.2298	20.862	1.6479.10 ⁻²	-9.7972	18.921	4.1732.10 ⁻¹
9	1.2301	15.864	-4.0925.10⁻³	16.5058	18.999	7.5058.10⁻³
10	1.2299	15.491	4.9065.10 ⁻³	18.4713	18.965	1.8683.10 ⁻¹
12	1.2292	17.886	6.7485.10 ⁻²	5.8637	18.728	1.4337
15	1.2301	14.645	-7.8094.10 ⁻³	22.9209	19.013	-6.6509.10 ⁻²
20	1.2306	13.510	-4.6183.10 ⁻²	28.8945	19.158	-8.3115.10 ⁻¹
22	1.2303	16.025	-2.1074.10 ⁻²	15.6586	19.063	-3.3096.10 ⁻¹

Tabla VII.2.17: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura 1 *hidden layer* y distintas Configuraciones, cuando ΔS_{23}^{\ddagger} es computado mediante AH.

A la vista de los resultados de la Tabla VII.2.17 se puede concluir que la aplicación del algoritmo AGDC para la optimización individual de ΔS_{23}^{\ddagger} conduce a una disminución considerable de la desviación con respecto los resultados obtenidos solamente mediante ANN. Este hecho permite concluir que el valor de ΔH_{23}^{\ddagger} obtenido mediante ANN posee una desviación pequeña con respecto al valor real y se puede considerar aceptable. Por el contrario

el valor de ΔS_{23}^\ddagger que nos proporciona ANN puede ser mejorado considerablemente cuando se aplica el algoritmo AGDC exclusivamente para la optimización de dicho parámetro.

Tal y como hemos hecho en todos los casos, por último hemos aplicado el Algoritmo Híbrido para determinar los valores de ΔH_{23}^\ddagger y ΔS_{23}^\ddagger en una serie de experiencias cinéticas no isotérmicas para las que su valor es desconocido. En la Tabla VII.2.18 se ofrecen los resultados obtenidos tras la aplicación del AH, en primer lugar se muestran los valores alcanzados tras el proceso de *Predicción* realizado con la ANN óptima y su correspondiente desviación. A continuación se muestran los valores de ΔH_{23}^\ddagger y ΔS_{23}^\ddagger optimizados mediante la aplicación de AGDC y sus correspondientes desviaciones.

(ANN		- AGDC)		AH	
Valores Reales (ΔH_{23}^\ddagger)/ 10^5 Jmol ⁻¹	Valores Reales (ΔS_{23}^\ddagger)/ JK ⁻¹ mol ⁻¹	Valores Optimizados (ΔH_{23}^\ddagger)/ 10^5 Jmol ⁻¹	Valores Optimizados (ΔS_{23}^\ddagger)/ JK ⁻¹ mol ⁻¹	% Des. (ΔH_{23}^\ddagger)	% Des. (ΔS_{23}^\ddagger)
1.3074	18.881	1.3063	19.220	-3.3413.10 ⁻¹	-7.4453
1.1461	20.272	1.1524	18.321	4.8714.10 ⁻¹	8.9013
1.2779	19.134	1.2794	18.691	-1.0954.10 ⁻¹	-2.3659
1.2533	19.470	1.2550	18.942	-8.0850.10 ⁻²	-1.6764
1.2055	20.151	1.2070	19.686	-8.3970.10 ⁻²	-1.6272
1.1801	20.412	1.1882	19.783	-9.4882.10 ⁻³	-2.1373.10 ⁻¹
1.2285	19.802	1.2294	19.542	5.2294.10 ⁻²	1.0103
1.2304	19.781	1.2315	19.449	-1.1935.10 ⁻¹	-2.3625

Tabla VII.2.18: Resultados de los 8 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (9.00.10⁵, 1, 2) con 1 *hidden layer* y Configuración 6, cuando ΔS_{23}^\ddagger y ΔH_{23}^\ddagger son computados después de la aplicación de AH (ANN-AGDC).

Al igual que en el caso anterior hemos optimizado exclusivamente ΔS_{23}^\ddagger mediante la aplicación del algoritmo AGDC. Los resultados se ofrecen en la Tabla VII.2.19 y permiten comprobar la disminución de la desviación en la mayoría de los casos si lo comparamos con los resultados obtenidos cuando se aplica ANN exclusivamente.

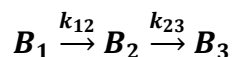
(ANN		- AGDC)
Valores Reales (ΔS_{23}^\ddagger)/ JK ⁻¹ mol ⁻¹	Valores Optimizados (ΔS_{23}^\ddagger)/ JK ⁻¹ mol ⁻¹	% Des. (ΔS_{23}^\ddagger)
18.881	19.220	-7.4453
20.272	18.321	8.9013
19.134	18.691	-2.3659
19.470	18.942	-1.6764
20.151	19.686	-1.6272
20.412	19.783	-2.1372.10 ⁻¹
19.802	19.542	1.0103
19.781	19.449	-2.3625

Tabla VII.2.19: Resultados de los 8 procesos de *Predicción* de la Red Neuronal constituida por una Arquitectura (900000, 1, 2) con 1 *hidden layer* y Configuración 9, cuando ΔS_{23}^\ddagger es computado después de la aplicación de AH (ANN+AGDC).

VII.3.- Determinación de constantes de velocidad mediante ANN

Predicción de constantes de velocidad mediante ANN

En el presente trabajo hemos analizado la aplicación de Redes Neuronales Artificiales para la determinación de constantes cinéticas (k_{12} y k_{23}) a partir de datos cinéticos, procedentes del mecanismo de reacción constituido por dos reacciones consecutivas irreversibles:



Al igual que en el caso de la determinación de PTA, en primer lugar se lleva a cabo el tratamiento de datos cinéticos sintéticos para comprobar la aplicabilidad de la metodología y posteriormente se realiza el tratamiento de datos sintéticos experimentales.

VII.3.1.- Tratamiento de datos sintéticos

i) Diseño de Experimentos

Cuando se pretenden determinar los valores de las constantes individuales de velocidad del mecanismo formado por dos reacciones consecutivas irreversibles, en el ED están involucrados dos *factores* (k_{12} y k_{23}), cuyas *respuestas* son los datos cinéticos de la base de curvas de *Inputs*. Estas respuestas pueden ser de 2 tipos según correspondan a la contribución de todas las especies (A_T) o a una sola especie (A_j , $[B_j]$ y α_j). En el ED de la Red Neuronal, es necesario tener en cuenta 2 variables: los valores límite de los dos *factores* que configuran el *dominio experimental* y sus valores relativos (k_{12} / k_{23}). Las combinaciones binarias de las constantes de velocidad generarán un conjunto de curvas cinéticas deben poseer información suficiente para garantizar un proceso de *Entrenamiento* óptimo de la Red Neuronal. Dentro del intervalo de tiempo considerado, deben cumplirse 2 requisitos: a) la cinética deberá alcanzar una conversión de al menos de 80% ($\xi'_{\text{end}} = 0.8\xi'_{\text{max}}$) y b) los valores de los datos cinéticos deben variar dentro de al menos 40% -50% para que la Red Neuronal sea capaz de discriminar claramente entre las diferentes curvas de *Inputs*. Además, el número de *niveles* de los *factores* del ED debe ser adecuado si se quiere evitar un trabajo computacional inútil y evitar grandes diferencias en las distancias de los valores de las *respuestas*. En consecuencia, para optimizar el proceso de *Entrenamiento* de la ANN, las curvas cinéticas de la matriz *Inputs* deben tener información cinética eficiente y deben ser distribuidas correctamente de acuerdo con la elección de un diseño experimental adecuado y un *Dominio Experimental* apropiado. Teniendo en cuenta las características del sistema cinético en estudio, hemos elegido como el más adecuado, el ED correspondiente a la distribución del tipo *Central Star Composite Experimental Desing* (CSCED) para la combinación con la técnica de Redes Neuronales Artificiales.

Para determinar el número óptimo de curvas se analizaron el número ideal de datos cinéticos para cada curva. Hemos probado con 25, 50, 75 y 100 valores, siendo el óptimo, el de 50 datos (50 neuronas), ya que la mejora en los resultados obtenidos con 75 y 100 valores no justifica el enorme aumento de tiempo de cálculo en cuestión. El rango máximo de variación

considerado, ha sido el mismo para ambos *factores*, con valores límite de $3.12 \cdot 10^{-2}$ y $1.19 \cdot 10^{-1} \text{ min}^{-1}$. Hemos variado el número de niveles y nodos de ambos *factores*, analizando las *respuestas* obtenidas. Se verificaron las configuraciones formadas por 29, 37, 45 y 53 nodos o curvas, siendo la configuración óptima aquella formada por 45 nodos, correspondientes a 9 niveles y 4 sub-niveles (Figura VII.3.1). Este valor ofreció un número suficiente de *respuestas*, sin haber grandes diferencias en el espaciamiento, y también es un número adecuado de curvas para el trabajo computacional que debe llevarse a cabo. Las dimensiones de la matriz de *Inputs* son 50×45 .

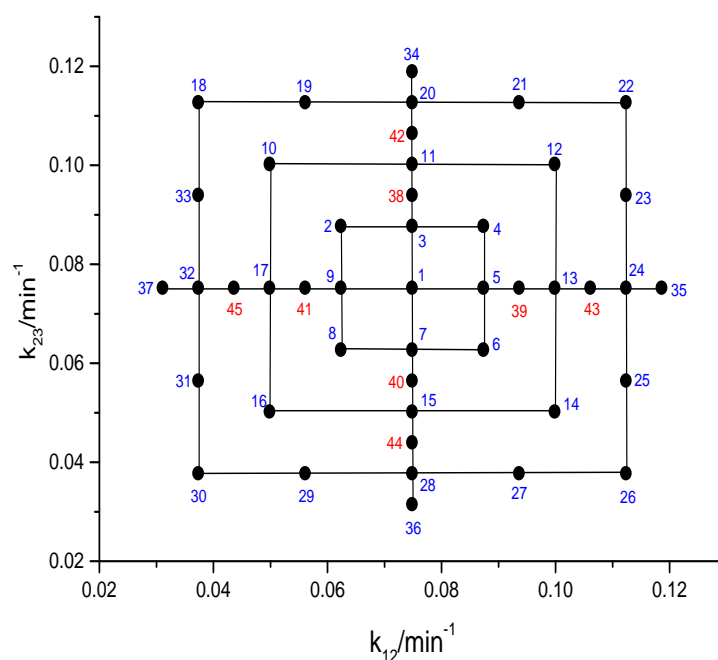


Figura VII.3.1: Diseño de Experimentos (CSCED) constituido por 2 factores (k_{12} y k_{23}) y 45 puntos distribuidos en 9 niveles (puntos del 1-37) y 4 subniveles (puntos del 38 - 45) que participan en la base de curvas de la matriz de *Inputs* utilizada para el proceso de *Entrenamiento* de la Red Neuronal.

ii) *Entrenamiento*

Con el fin de determinar la Arquitectura óptima de la Red Neuronal se evaluó la influencia de las diferentes variables en el proceso de *Entrenamiento* de la ANN, evaluando el efecto de cada uno de ellos en cada capa.

Input layer: Dado que las curvas cinéticas fueron generadas sintéticamente con las funciones explícitas a partir del conjunto de valores de los parámetros cinéticos (k_{12} y k_{23}), estructurados de acuerdo al Diseño de Experimentos (ED) anterior, los porcentajes en los procesos de *Validation* y *Testing* pueden ser altos con respecto a las curvas de *Training*. Este proceso se realiza con una distribución de las curvas para *Training*, *Validation* y *Testing* en porcentajes 80/10/10. A estos datos cinéticos les imponemos distintos valores o niveles de *noise*, los cuales son de una magnitud igual a la de los errores experimentales, esto es, en el caso

de las concentraciones: $\pm 1.0 \times 10^{-8}$ ($[B_1]_0 = 5.50 \cdot 10^{-5} \text{ mol dm}^{-3}$) y en el caso de las absorbancias: $\pm 1 \times 10^{-4}$.

Hidden layer/s: La determinación del número óptimo de capas Ocultas de cada una de ellas es crucial para la determinación de la Arquitectura óptima de la Red Neuronal, de manera que es necesario probar el mayor número de casos posibles. El procedimiento, del tipo ensayo y error, consistirá en fijar inicialmente el número de capa/s Ocultas, y llevar a cabo el *Entrenamiento* de la Red con diferentes Configuraciones. Tendremos en cuenta aquellos resultados obtenidos para las Configuraciones más significativas y que conducen a un valor de MSE más pequeño. La determinación de los valores mínimos de MSE se lleva a cabo representando los valores de MSE vs número de neuronas para cada Configuración. Como se explicó en capítulos anteriores, el mínimo observado o el punto de intersección de las dos ramas, siempre (en abscisas) es el número óptimo de neuronas en la capa oculta.

Para los datos cinéticos de absorbancia total/tiempo, y considerando en primer lugar una ANN constituida por una sola capa Oculta, se obtiene una Configuración óptima de 5 nodos que corresponde al mínimo valor de MSE. Cuando tomamos una Arquitectura con 2 capas Ocultas y diferentes Configuraciones para ambas, en este caso, la Configuración que nos conduce a un valor de MSE más bajo es 6/10, obteniendo una estructura multicapa que mejora los resultados frente a aquellos logrados por una sola capa Oculta.

Output layer: El número de neuronas de la capa de salida corresponde al número de parámetros a predecir que en nuestro caso es 2. Hemos llevado a cabo una serie de experiencias computacionales sintéticas en las que se realiza el *Entrenamiento* utilizando los diferentes algoritmos que proporciona MATLAB a través de la aplicación *Neural Network Toolbox*, seleccionando como el mejor, aquel que nos proporciona los mejores resultados. Evaluamos los obtenidos al aplicar los métodos *Steepest Descent Method* (SDM), *Técnicas Heurísticas*, *Gradiente Conjugado*, *Quasi-Newton*, *Algoritmo Levenberg-Marquardt* (LM), etc. Después del análisis realizado, observamos una clara diferencia entre los resultados obtenidos con el *Algoritmo Levenberg-Marquardt* y los correspondientes a los restantes métodos. Los valores obtenidos para el error cuadrático medio (MSE) y los restantes parámetros que se obtienen tras el proceso de *Entrenamiento*, nos permiten concluir que el algoritmo que proporciona los mejores resultados es el *Levenberg-Marquardt*. Por tanto, éste es el elegido para ser utilizado en los tratamientos ANN para estos sistemas.

De acuerdo, con todo lo comentado anteriormente, el proceso de *Entrenamiento* de la ANN lo llevamos a cabo generando datos sintéticos dotados de un *noise* del orden de los errores experimentales, sobre la base de la Arquitectura óptima que hemos obtenido, con la siguiente estructura: las capas de entrada y salida tienen 50 y 2 neuronas respectivamente, mientras que la composición de la/s capa/s Oculta/s dependía del tipo de datos cinéticos de la base de curvas. En el caso de los datos individuales (A_j , $[B_j]$ y α_j), la Arquitectura óptima resultó ser (50, 1, 2) con

una sola capa Oculta, cuya Configuración adecuada tiene 5 nodos. Y para los datos de A_T , la Arquitectura óptima es (50, 2, 2) con 2 capas Ocultas y Configuración 6/10.

El proceso de *Entrenamiento* de la Red Neuronal óptima lo realizamos y posee las siguientes características: a) los datos cinéticos de la matriz de *Inputs*, forman una base de curvas compuesta por 45 curvas, de acuerdo con el CSCED elegido, con 50 datos para cada curva y una organización estándar en el caso de A_T , y secuencial para los casos A_j , $[B_j]$ y α_j ; b) el valor de *noise* impuesto a los datos sintéticos $\pm 1 \times 10^{-4}$ para los datos de absorbancia y $\pm 1.0 \times 10^{-8}$ para los datos de la concentración; c) en todas las curvas que consideramos los datos cinéticos poseen una conversión máxima de no menos del 82 % ($\xi'_{\text{end}} = 0.82\xi'_{\text{max}}$); d) los valores de los porcentajes de *Training*, *Validation*, y *Testing* elegidos en todos los casos son 80/10/10; y e) el Algoritmo de optimización más adecuado es el de *Levenverg-Marquardt* (LM).

Los perfiles de las líneas de Regresión *output/target*, se muestran en la Figura VII.3.2, correspondiente a los procesos de (a) *Training*, (b) *Validation*, (c) *Testing* y (d) conjunto de todos los procesos al mismo tiempo. Los resultados obtenidos conducen a valores satisfactorios de MSE y órdenes de magnitud similares al *noise* impuesto a los datos. La distribución de los residuales muestra fluctuaciones aceptables y las Regresiones tienen ecuaciones con pendientes cercanas a la unidad, con ordenadas prácticamente nulas en el origen, y muy buenos valores (0.99998) de los coeficientes de regresión (R).

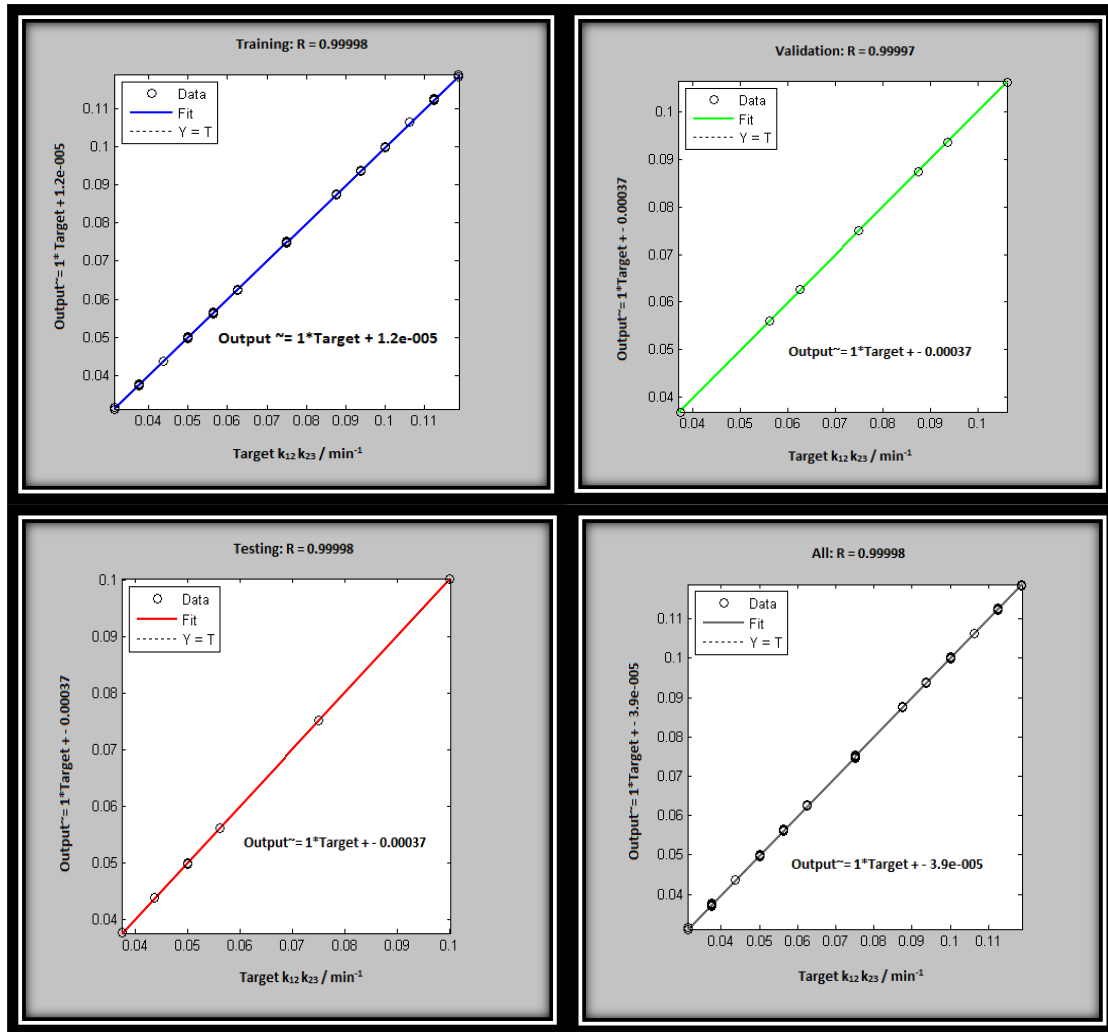


Figura VII.3.2: Representación de las Regresiones *Outputs/Targets* del conjunto de curvas, para los procesos de *Training*, *Validation*, *Testing* y representación conjunta para tres procesos anteriores.

iii) *Predicción*

El proceso de *Predicción* de las constantes cinéticas a partir de datos sintéticos y puesto que los valores de k_{12} y k_{23} son conocidos, nos permite evaluar la influencia de una serie de variables como son el valor de *noise*, el número de curvar a predecir y el número mínimo de especies necesarias para llevar a cabo el proceso de *Predicción*.

a) **Valor del *noise*:** Se ha analizado la influencia que tiene en los resultados el *noise* o error al azar impuesto a los datos sintéticos con objeto de obtener la magnitud óptima de *noise* de los datos cinéticos. El análisis de esta influencia nos permite conocer la precisión del método ANN puesto que, con los valores conocidos de las constantes cinéticas que generan los datos y los determinados por el método ANN, podemos calcular su error o Desviación Estándar (SD). El procedimiento que hemos seguido consiste en generar sintéticamente datos cinéticos a partir de un conjunto de valores de k_{12} y k_{23} a los que se le imponen distintos valores o niveles de *noise*, tanto en el caso de la matriz de *Inputs* para configurar la base de curvas de

Entrenamiento de la Red Neuronal como para las de *Predicción*. Con los valores de las k cinéticas obtenidos de la *Predicción* y los valores de partida que sirvieron para generar los datos, determinamos los valores de la Desviación Estándar (SD) para ambas constantes cinéticas (k_{12} y k_{23}). La influencia que el *noise* ejerce sobre los errores o SD de las constantes cinéticas se puede observar realizando la representación gráfica de los valores obtenidos de SD frente a sus correspondientes valores de *noise*. No se pretende realizar una correlación rigurosa es suficiente conocer los valores individuales de SD y los límites de los *factores* del *dominio experimental* en el rango máximo del ED. Estos valores son:

$$k_{12} = (3.1300 \cdot 10^{-2} \pm 3.6000 \cdot 10^{-4}) \text{ min}^{-1} \text{ y } k_{23} = (1.1880 \cdot 10^{-2} \pm 2.5000 \cdot 10^{-4}) \text{ min}^{-1}$$

$$k_{12} = (1.1880 \cdot 10^{-2} \pm 3.6000 \cdot 10^{-4}) \text{ min}^{-1} \text{ y } k_{23} = (1.1880 \cdot 10^{-2} \pm 2.5000 \cdot 10^{-4}) \text{ min}^{-1}$$

b) Número de curvas *Input*: Determinamos el número óptimo de curvas para el proceso de *Predicción* generando datos sintéticos de A_T , A_j , $[B_j]$ y α_j , a partir de valores de las constantes (k_{12} , k_{23}) procedentes del ED y no coincidentes con aquellos utilizados en el proceso de *Entrenamiento*, como son los correspondientes a los puntos 1-16 de la Figura VII.3.3. La Figura VII.3.4 muestra la representación gráfica de los valores de SD para ambas constantes de velocidad, determinadas en los procesos de *Predicción*, con respecto a la variación del número de curvas. Puede observarse que, en general, al aumentar moderadamente el número de curvas moderadamente, el valor de SD disminuye para ambas constantes de velocidad a un determinado valor umbral, después del cual los valores de SD muestran una tendencia a disminuir ligeramente o a mantenerse constantes. Este valor resultó ser el óptimo en el rango de curvas de 8-12 cuando partimos de una base de *Entrenamiento* de 45 curvas *Inputs*.

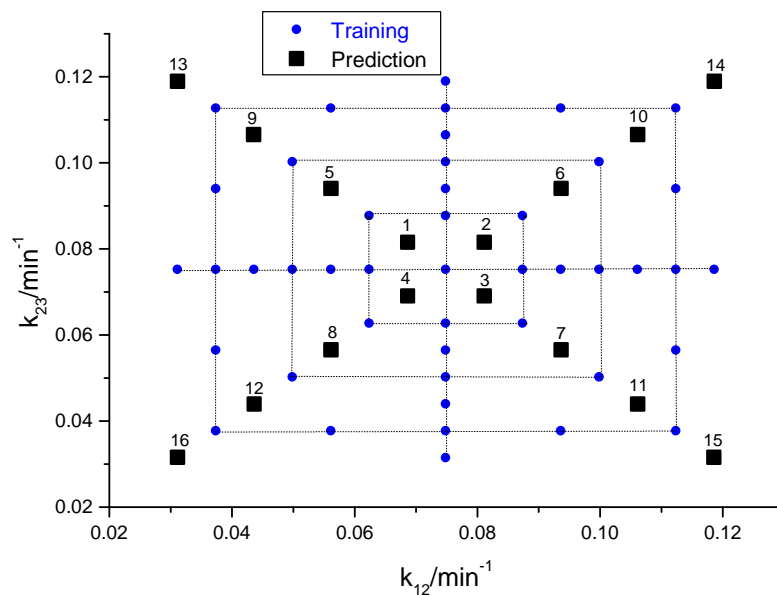


Figura VII.3.3: Representación gráfica del Diseño de Experimentos (CSCED) donde los 16 puntos para la *Predicción* (1-16) se representan de forma conjunta con los 45 puntos utilizados para el proceso de *Entrenamiento* de la Red Neuronal.

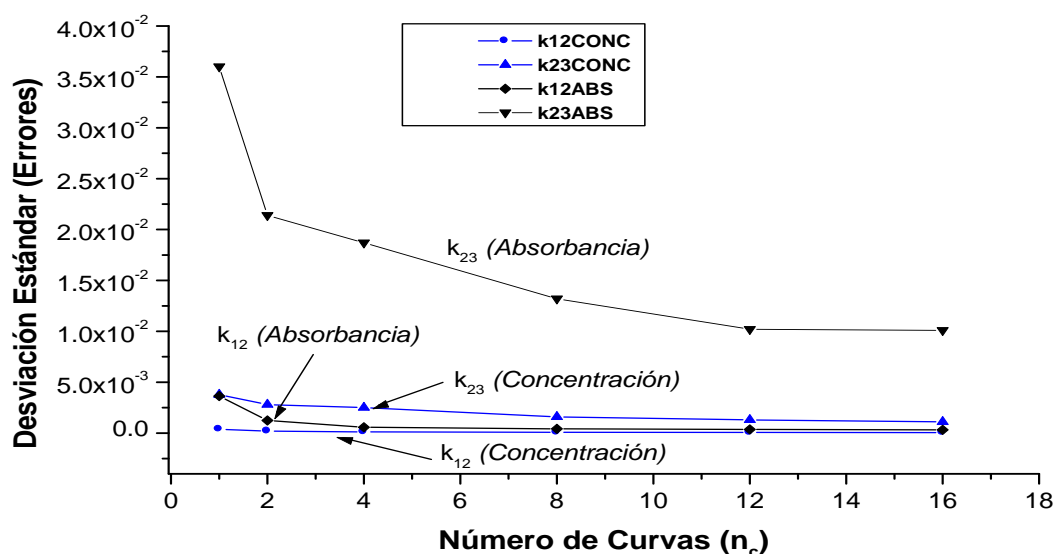


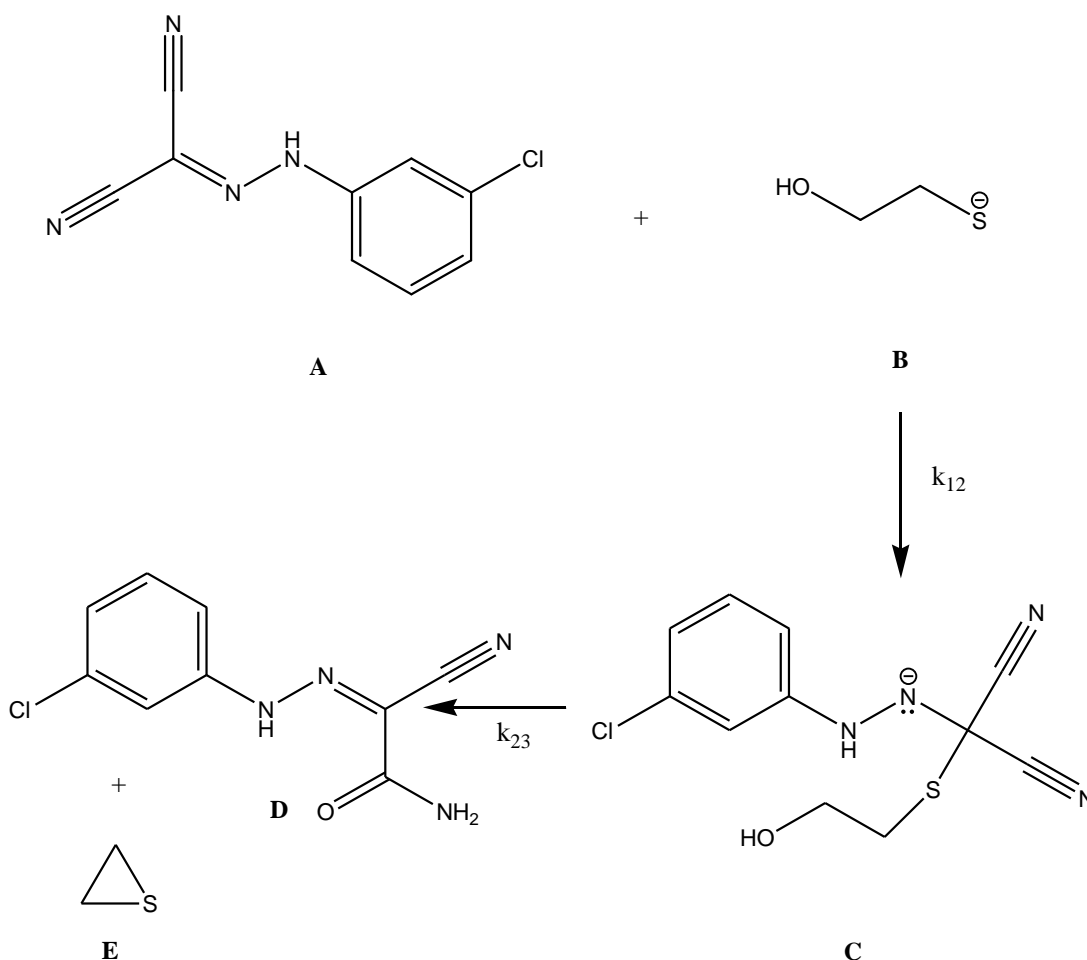
Figura VII.3.4: Representación gráfica de los valores de SD obtenidos en el proceso de *Predicción* de ambas constantes de velocidad (k_{12} y k_{23}) vs el número de curvas cuando calculamos tanto los datos de absorción como de concentración.

c) **Número mínimo de especies:** Para determinar el número de especies cuya contribución a los datos cinéticos es necesario tener en cuenta para realizar la *Predicción*, tenemos que analizar los resultados obtenidos cuando los procesos de *Entrenamiento* y *Predicción* se realizan a partir de la computación de datos cinéticos que incluyen la contribución de una única especie, cuando lo hacen 2 especies o bien mediante computación de los datos que aportan las 3 especies. Esta situación es bastante común, cuando en el espectro no todas las especies absorben o cuando la absorción de una de ellas predomina sobre las otras dos ya que los coeficientes de absorción molar son muy diferentes. Cuando se computan datos cinéticos de absorancias que incluyen la contribución de 2 especies, puede completarse perfectamente el proceso de *Predicción* mediante ANN alcanzándose valores idénticos de k_{12} y k_{23} a los obtenidos mediante computación ANN de los datos que aportan las 3 especies. Si se computan los datos cinéticos de la especie intermedia (B_2), que es la que está involucrada en las 2 reacciones consecutivas, se pueden *Prededir* con éxito ambas constantes (k_{12} y k_{23}). Lo que parece seguro, es que al menos una de las 2 constantes cinéticas puede obtenerse con la computación ANN de una sola especie.

VII.3.2. – Tratamiento de datos experimentales

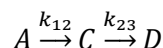
Una vez que ha sido comprobada la aplicabilidad de la metodología ANN para la predicción de constantes de velocidad (a partir de datos cinéticos sintéticos) para el mecanismo de reacción de dos reacciones consecutivas irreversibles, se aplica dicha metodología para el

tratamiento de datos experimentales. Hemos elegido una reacción que responde al mecanismo considerado y cuya cinética ha sido ampliamente estudiada por diversos y prestigiosos autores, que obtienen resultados concordantes lo cual garantiza su fiabilidad. La reacción seleccionada, que cumple todos estos requisitos, es la que tiene lugar entre el 2-mercaptoetanol con Carbonil Cianida 3-Clorofenilhidrazona ((3-Cl)-PHPD) que es un potente “desacoplador” de la fosforilación oxidativa en las células. La reacción del (3-Cl)-PHPD (*A*) y compuestos similares con 2-mercaptoetanol (*B*) ha sido estudiada por diferentes autores ⁽¹⁵⁸⁻¹⁶⁰⁾ los cuales consideran en sus trabajos que en un primer paso se forma un aducto intermedio (*C*) que posteriormente se hidroliza en una reacción aparentemente intramolecular, para dar el producto Cianoacetamida 3-Clorofenilhidrazona ((3-Cl)-PHCA), (*D*) y el subproducto sulfuro de etileno (*E*). La reacción transcurre de la siguiente manera:

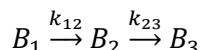


Esquema VII.3.1: Reacción consecutiva entre 2-mercaptoetanol y Carbonil Cianida 3-Clorofenilhidrazona.

Cuando la reacción se lleva a cabo en exceso de *B* se puede suponer pseudo-primer orden y por tanto:



que en general y para adaptarnos a la nomenclatura IUPAC⁽¹⁴⁶⁻¹⁴⁷⁾, formulamos:



donde la especie A es B_1 , la C es B_2 y la D es B_3 .

Chau y col.⁽¹⁵⁹⁾ realizaron un estudio cinético clásico de la reacción proporcionando datos experimentales de absorbancia total/tiempo obtenidos a las diferentes longitudes de onda (350, 375, 400 y 450 nm), cuando la reacción se lleva a cabo a partir de unas determinadas condiciones de temperatura (20 °C) y de pH (4.3). La evolución de la reacción se monitoriza midiendo la absorbancia total de la mezcla reaccionante durante 30 minutos. En la Figura VII.3.5 se muestra el progreso de la reacción cuando se monitoriza la absorbancia medida a 350, 375, 400 y 450 nm.

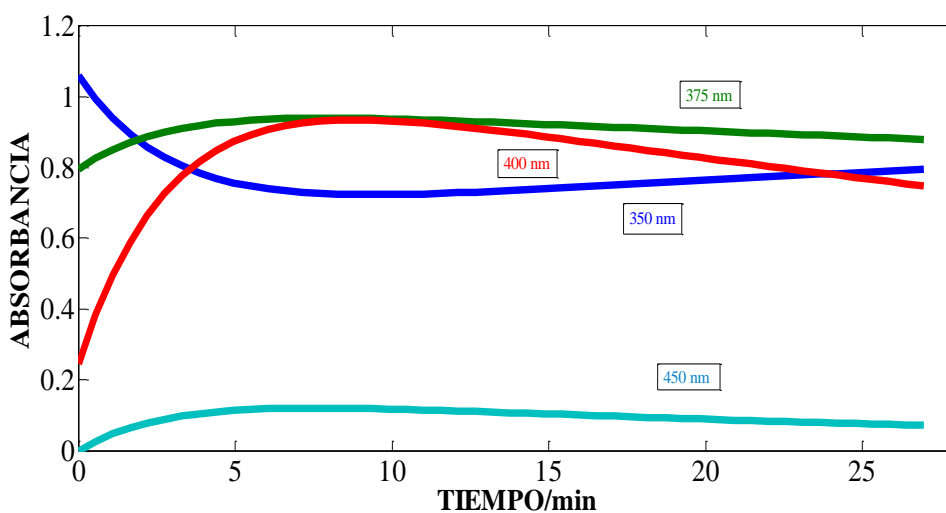


Figura VII.3.5: Representación gráfica del progreso de la reacción durante 30 min cuando se monitoriza la absorbancia medida a distintas longitudes de onda.

El tratamiento de los datos experimentales mediante procedimientos clásicos de Regresión no lineal permitió a estos autores determinar las constantes de velocidad y los coeficientes de absorción molar de cada especie a cada longitud de onda. Los resultados obtenidos están indicados en la Tabla VII.3.1.

λ /nm	k_{12} /min ⁻¹	k_{23} /min ⁻¹	$\epsilon_{B_1}^\lambda \times 10^{-4}$ /mol ⁻¹ cm ⁻¹ L	$\epsilon_{B_2}^\lambda \times 10^{-4}$ /mol ⁻¹ cm ⁻¹ L	$\epsilon_{B_3}^\lambda \times 10^{-4}$ /mol ⁻¹ cm ⁻¹ L
350	0.33360	0.021000	2.1700	1.3600	2.0400
375	0.33600	0.020040	1.6300	2.0000	1.4800
400	0.33360	0.020280	0.50000	2.2000	0.46000
450	0.34200	0.032400	0	0.31000	0

Tabla VII.3.1: Valores de las constantes de velocidad y los coeficientes de absorción molar de cada especie a cada longitud de onda.

En vista de la concordancia entre los resultados obtenidos por estos autores, cuando la reacción se controla a diferentes longitudes de onda, con los obtenidos en otras aportaciones bibliográficas, utilizamos estos datos experimentales y los resultados proporcionados en estos trabajos para la determinación de las constantes cinéticas (*Predicción*) para el mecanismo considerado. En nuestro caso, realizamos el tratamiento de los datos cinéticos experimentales de absorbancia/tiempo, aplicando técnicas de ANN dentro del algoritmo computacional híbrido AH.

i) **Diseño de Experimentos**

Para realizar la *Predicción* mediante ANN es necesario someter a la Red Neuronal Artificial a un proceso de *Entrenamiento*, para lo cual es necesario realizar el correspondiente Diseño de Experimentos con objeto de obtener las matrices *Inputs* y *Targets*. En este caso al tratarse de datos experimentales desconocemos el valor de las constantes de velocidad, incluso su orden de magnitud, por lo que no es posible realizar el correspondiente Diseño de Experimentos salvo que tengamos una estimación previa o algún indicio de los valores de las constantes de velocidad que nos proporcionen al menos sus órdenes de magnitud. Para obtener estos valores, a partir de los datos cinéticos experimentales de los que disponemos, hemos utilizado los siguientes procedimientos: Método Diferencial (Velocidades iniciales): Vida media y Relaciones en el máximo de concentración de la especie B_2 y en el punto de inflexión de B_3 . Este tratamiento previo nos proporciona los siguientes valores estimados de las constantes de velocidad: $k_{12} = 0.33000 \text{ min}^{-1}$ y $k_{23} = 0.02100 \text{ min}^{-1}$. En vista de estos valores establecemos el intervalo de variación para k_{12} y k_{23} con el que posteriormente hemos realizado el ED, que posee un intervalo del *Dominio Experimental* con valores extremos 0.35000-0.020000. De acuerdo a este intervalo, que hemos considerado idóneo, hemos realizado el Diseño de Experimentos, del tipo *Central Star Composite Experimental Desing* (CSCED), que nos ha servido para generar los datos destinados al *Entrenamiento* de la Red Neuronal.

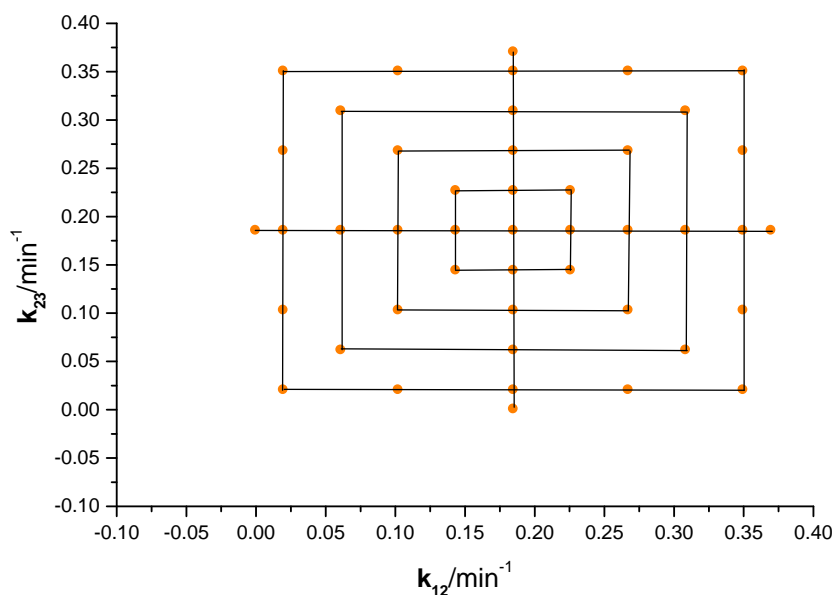


Figura VII.3.6: Representación del ED con los valores correspondientes al intervalo del *Dominio Experimental* 0.35-0.02.

ii) *Entrenamiento*

Una vez realizado el ED y generadas las curvas cinéticas, hemos llevado a cabo el proceso de *Entrenamiento* de la Red Neuronal Artificial (ANN) con el fin de encontrar la más adecuada que nos permita realizar la *Predicción* de las constantes de velocidad de manera satisfactoria. Al igual que hemos hecho en el caso del tratamiento de datos sintéticos, pretendemos encontrar la Red con Arquitectura, Estructura (número de capas) y Configuración (número idóneo de nodos) que resulte óptima para el tratamiento de los datos experimentales procedentes de la reacción objeto de estudio.

El *Entrenamiento* lo hemos llevado a cabo con una ANN formada por 2 capas ocultas y distintas Configuraciones, para todas las longitudes de onda de trabajo. La Tabla VII.3.2 muestra los resultados obtenidos en el proceso de *Entrenamiento* llevado a cabo mediante las 45 curvas cinéticas obtenidas a $\lambda = 350$ nm, $\lambda = 375$ nm, $\lambda = 400$ nm y $\lambda = 450$ nm., a partir del Diseño de Experimentos, considerando una Red cuya Arquitectura consta de 2 capas ocultas (50, 2, 2) y en las que se varía las Configuraciones de las capas Ocultas. La Tabla VII.3.2 contiene los valores de MSE obtenidos para el ED, con aquellas Configuraciones (3/8, 5/7, 4/10, 5/10, 6/10), que proporcionan un menor valor de MSE de *Validación*. En los resultados que aparecen en la Tabla 2 se observa que para el caso de $\lambda = 350$ nm la Configuración 5/7 es la que conduce a un valor menor de MSE y es la que elegimos para realizar la *Predicción* a 350 nm. En cambio, cuando los datos de absorbancia se obtienen a las longitudes de onda de 375, 400 y

450 nm., la Configuración 6/10 es la que conduce a valores menores de MSE, y es la elegida para estas 3 λ .

λ/nm	350	375	400	450
N1/N2	MSE			
3/8	$6.3701 \cdot 10^{-7}$	$2.4005 \cdot 10^{-8}$	$1.2002 \cdot 10^{-7}$	$2.4404 \cdot 10^{-7}$
5/7	$2.5802 \cdot 10^{-10}$	$4.8004 \cdot 10^{-9}$	$1.8503 \cdot 10^{-7}$	$1.0401 \cdot 10^{-8}$
4/10	$8.0003 \cdot 10^{-8}$	$1.7203 \cdot 10^{-9}$	$1.2401 \cdot 10^{-7}$	$8.0502 \cdot 10^{-9}$
5/10	$2.6304 \cdot 10^{-9}$	$3.4102 \cdot 10^{-9}$	$2.2504 \cdot 10^{-8}$	$8.2205 \cdot 10^{-9}$
6/10	$4.3105 \cdot 10^{-9}$	$1.4201 \cdot 10^{-11}$	$5.2305 \cdot 10^{-9}$	$4.8903 \cdot 10^{-11}$

Tabla VII.3.2 Valores de MSE obtenidos para el ED con algunas Configuraciones durante el proceso de *Entrenamiento*.

iii) *Predicción*

El proceso de *Predicción* de los parámetros cinéticos desconocidos (constantes de velocidad), ha consistido en utilizar de la bibliografía datos experimentales de garantía suficientemente contrastada de k_{12} y k_{23} , correspondientes a una reacción química real que tiene lugar en el laboratorio. Con los valores experimentales de las constantes cinéticas, coeficientes de absorción molar de las 3 especies a las 4 longitudes de onda de trabajo, concentraciones iniciales y las condiciones experimentales de reacción descritas en la bibliografía, obtenemos datos de absorbancias a distintos valores de tiempo a las 4 λ de trabajo (350, 375, 400 y 450 nm.) a los que se les impone como *noise* el error experimental. Estos datos los hemos sometido al proceso de *Predicción* con la Red Neuronal óptima que resulta del *Entrenamiento*, para determinar (*predecir*) los valores desconocidos de k_{12} y k_{23} . En todos los casos, hemos realizado el correspondiente Análisis de Residuales que nos ha permitido determinar el Error cuadrático del ajuste (*SD*) de acuerdo a la ecuación [VII.1.2].

$$SD = \left(\frac{\sum_{i=1}^{n_d} \sum_{j=1}^{n_c} (b_{ij}^{exp} - b_{ij}^{calc})^2}{n_d \cdot n_c} \right)^{1/2} \quad \text{[VII.1.2]}$$

donde b_{ij}^{exp} corresponde a los valores de absorbancia total de entrada de la matriz de *Inputs*; b_{ij}^{calc} representa los valores de absorbancia total calculados con los valores de k_{12} y k_{23} que han sido determinados tras el proceso de *Predicción*; n_d el número de valores de absorbancia de cada curva y n_c el número de curvas, que en este caso particular es la unidad ($n_c = 1$) para cada λ .

En la Tabla VII.3.3 se encuentran los resultados conseguidos en los procesos de *Predicción* de las constantes de velocidad obtenidos a partir de los datos experimentales a cada λ (350, 375, 400 y 450 nm.). La 2ª y 3ª columnas muestran los valores de las constantes de velocidad determinados mediante *Predicción* ANN y en la 4ª columna se encuentran recogidos los valores de la Desviación Estándar (*SD*) calculados mediante la ecuación [VII.1.2].

λ/nm	k_{12}/min^{-1}	k_{23}/min^{-1}	SD
350	0.33320	0.02043	$9.2621 \cdot 10^{-4}$
375	0.33591	0.02022	$7.7012 \cdot 10^{-3}$
400	0.34542	0.01791	$6.1040 \cdot 10^{-3}$
450	0.36723	0.03180	$5.9223 \cdot 10^{-4}$

Tabla VII.3.3: Resultados conseguidos en los procesos de *Predicción* de las constantes de velocidad.

VII.4.- Discriminación de mecanismos de reacción

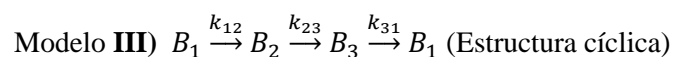
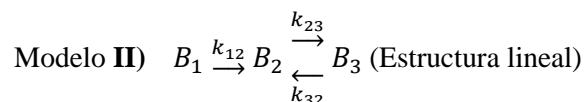
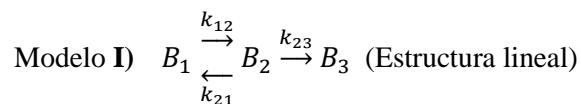
VII.4.1.- Discriminación de mecanismos de reacción mediante ANN

VII.4.1.1.- Capacidad de Discriminación de mecanismos en modelos “Soft Modelling”

Probablemente una de las posibilidades más importantes de aplicación de la técnica ANN en sistemas cinéticos, es cuando se desarrolla bajo la modalidad *Soft Modelling*. El método *Soft Modelling* añade a la determinación clásica de parámetros cinéticos y termodinámicos de activación, la valiosa posibilidad de establecer el modelo más probable al que los datos cinéticos pertenecen experimentales, ya que la metodología cinética apunta la capacidad de discriminación entre los distintos mecanismos o modelos que se consideran aplicables.

En el presente trabajo, hemos realizado una serie de pruebas para analizar la aplicabilidad de la técnica ANN para discriminar entre modelos en Cinética Química que proporciona buenos resultados cuando actúa en la modalidad cinética tipo *Soft modelling*. Hemos analizado una de las posibilidades, como más significativa e importante:

- Discriminación entre 3 modelos distintos en el que intervienen el mismo número de especies, de reacciones entre ellas y de constantes de velocidad. Los modelos son diferentes en cuanto a su arquitectura (lineal o cíclica) o bien en cuanto a la distribución de las reacciones individuales o constantes cinéticas. Ejemplo: (Modelos con 3 especies y 3 constantes cinéticas)

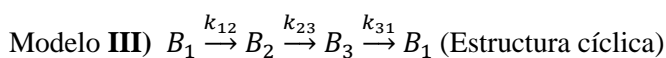
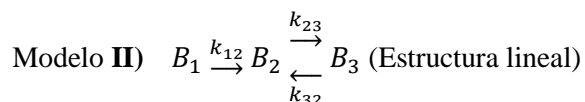
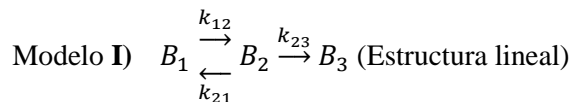


Vamos a comentar el procedimiento que hemos utilizado en el tratamiento, ya que es el que consideramos posee una mayor importancia. Elegimos “*a priori*” un modelo base (p. ej. el modelo **I**) de los 3 que pretendemos discriminar mediante la aplicación de técnicas ANN.

Generamos los datos cinéticos (50 valores de absorbancias totales de la mezcla) a partir únicamente del modelo base (I) con los grupos de sus 3 constantes de velocidad, organizadas de acuerdo a un Diseño de Experimentos tipo *Central Star Composite Experimental Design*, (CSCED). Esto nos proporciona un colectivo de 59 curvas cinéticas, a cuyos datos, les asignamos los correspondientes valores apropiados del error al azar (*noise*) de un orden de magnitud igual al experimental ($\pm 1.10^{-4}$). Estos datos conforman la matriz de *Inputs* (dimensiones 59x50) y conjuntamente con la matriz de *Targets* (59x3), se las suministramos a la Red Neuronal para proceder a su *Entrenamiento*. Una vez realizado satisfactoriamente este proceso, generamos para todos y cada uno de los 3 modelos (I, II y III), 59 curvas cinéticas, con 50 valores de absorbancias cada uno y ternas de valores de k cinéticas distintas a las que sirvieron para generar las curvas utilizadas en el *Entrenamiento*. Una vez que les imponemos los correspondientes valores de *noise* ($\pm 1.10^{-4}$), formamos 3 matrices de *Inputs*, con dimensiones de 59x50, correspondientes a cada uno de los 3 modelos (I, II y III) destinadas al proceso de *Predicción* individual de cada una de ellas por separado. Para ello, suministramos a la Red Neuronal ya entrenada exclusivamente para el modelo a1, la matriz de *Inputs* de cada uno de los 3 modelos proporcionándonos en cada caso la matriz de *outputs [Y]* (dimensiones 59x3), que contienen los valores de las constantes cinéticas que la Red Neuronal *Predice*, para cada modelo, correspondientes a las 59 curvas generadas al efecto. Pues bien, analizando estas 3 matrices *[Y]*, observamos que únicamente en la que corresponde al proceso de *Predicción* del modelo base (I), existe una perfecta coincidencia entre los valores de las constantes cinéticas con los que sirvieron para generar las 59 curvas. Sin embargo, en las otras 2 matrices *[Y]* correspondientes a los modelos II) y III), no solo es que no haya concordancia sino que se obtienen valores absurdos e inaceptables (incluso valores negativos de las k cinéticas) que ponen de manifiesto el fracaso en la *Predicción* de los valores de constantes cinéticas para estos 2 modelos con la *entrenada* exclusivamente para el caso I). A la vista de los resultados obtenidos, podemos concluir que la Red Neuronal (ANN) es capaz de distinguir entre modelos diferentes en el proceso de *Predicción* conduciendo a valores correctos de constantes de velocidad únicamente cuando se le suministran los datos cinéticos de *Input* pertenecientes al modelo que sirvió para realizar su *Entrenamiento*. En caso contrario, cuando no pertenece a dicho modelo, el proceso de *Predicción* fracasa, obteniendo valores absurdos e inaceptables. De igual forma si elegimos “*a priori*” otro modelo distinto al I), (por ejemplo II) o III)) y generamos los datos cinéticos vuelve a suceder lo mismo que lo explicado anteriormente comentado, obteniendo unos resultados óptimos para el sistema a partir del cual hemos generado los datos y unos valores absurdos e inaceptables para los modelos contrarios a los cuales queremos realizar la discriminación.

VII.4.1.2.- Resultados de la discriminación mediante ANN

Hemos llevado a cabo la discriminación entre 3 modelos distintos de reacción en los que intervienen el mismo número de especies, tienen lugar el mismo número de reacciones entre ellas y por tanto constan del mismo número de constantes de velocidad. Los modelos son diferentes en cuanto a su arquitectura (lineal o cíclica) o bien en cuanto a la distribución de las reacciones individuales o constantes cinéticas.



VII.4.1.2.1.- Modelo I

En el primer caso tomamos como base el **I**) generando una serie de curvas cinéticas para este mecanismo mediante la resolución numérica del sistema de ecuaciones diferenciales de velocidad (ODE). Para ello, utilizamos grupos de 3 constantes de velocidad, organizadas de acuerdo a un Diseño de Experimentos tipo *Central Star Composite Experimental Design*, (CSCED). El conjunto de ternas de constantes de velocidad elegidos para el punto central del ED es el siguiente $k_{12} = 0.07 \text{ min}^{-1}$, $k_{21} = 0.06 \text{ min}^{-1}$ y $k_{23} = 0.05 \text{ min}^{-1}$ (Figura VII.4.1) con sus correspondientes coeficientes de absorción molar, cuyos valores son $\varepsilon_1 = 20000$, $\varepsilon_2 = 18000$, $\varepsilon_3 = 15000$. Este ED es válido para todos los sistemas y es obviamente el que hemos utilizado para generar los datos de los 3 modelos elegidos para realizar la discriminación. En este caso concreto el espacio ED es un modelo en 3 dimensiones, ya que tenemos 3 parámetros a determinar, es decir, cada una de las constantes de velocidad que intervienen en cada mecanismo de reacción considerado. A partir de estos conjuntos de constantes hemos generado un colectivo de 59 curvas cinéticas, a cuyos datos de absorbancia les asignamos un error al azar (*noise*) de un orden de magnitud igual al experimental ($\pm 1.10^{-4}$). Estos datos conforman la matriz de *Inputs* (dimensiones 59x50) y conjuntamente con la matriz de *Targets* (59x3), se las suministramos a la Red Neuronal para proceder a realizar su *Entrenamiento*.

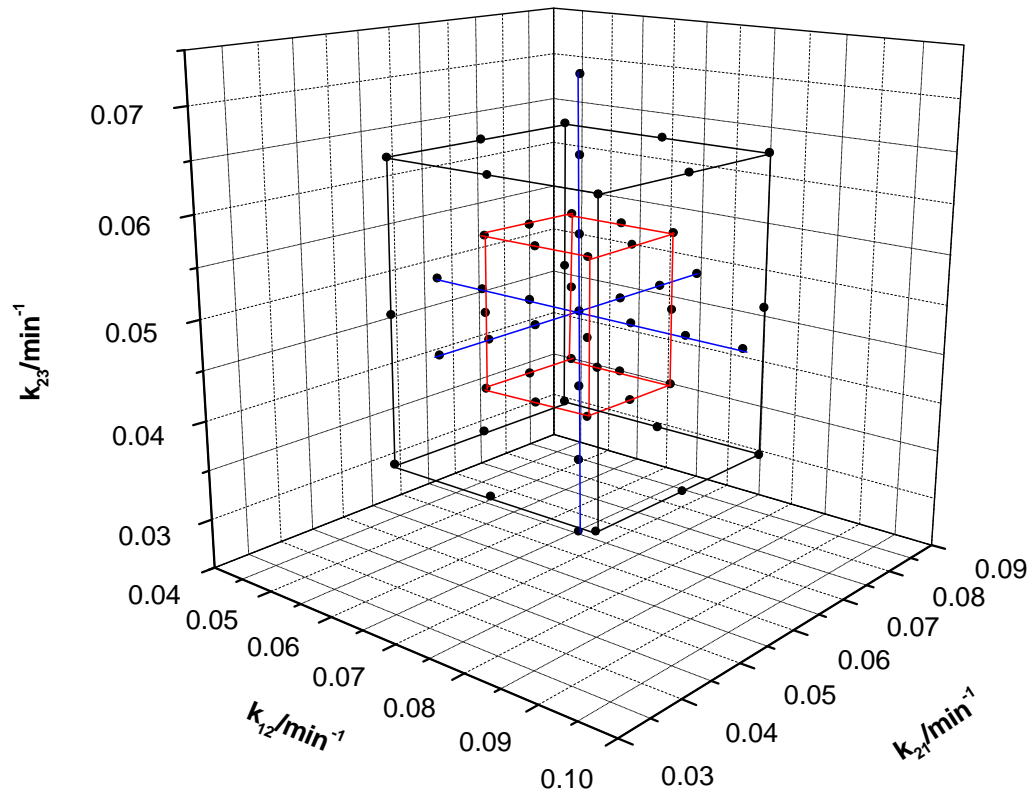


Figura VII.4.1: Diseño de Experimentos (CSCED) constituido por 3 factores (k_{12} , k_{21} y k_{23}) y 59 ternas de puntos que participan en la base de curvas de la matriz de *Inputs* utilizada para el proceso de *Entrenamiento* de la Red Neuronal.

Con el fin de determinar la Arquitectura óptima de la Red Neuronal se evaluó la influencia de las diferentes variables en el proceso de *Entrenamiento* de la ANN tal y como hemos explicado en el Capítulo VI. La Tabla VII.4.1 muestra los resultados obtenidos para los procesos de *Entrenamiento* de la ANN realizados con diferentes Estructuras (número de capas) y Configuraciones (número idóneo de nodos). En este caso la Arquitectura óptima consta de una Configuración de 3 capas ocultas y el número de nodos en cada capa es 9/10/10. Cuando se utiliza ANN para la determinación de constantes de velocidad podemos observar a simple vista que los resultados obtenidos son mucho mejores que cuando se determinan PTA, debido a la similitud en los órdenes de magnitud de las constantes.

ANN		Artificial		Neural	Networks	
Config.	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
7/10	0.069957	0.060515	0.050032	$6.1259 \cdot 10^{-2}$	$-8.5852 \cdot 10^{-1}$	$-6.4063 \cdot 10^{-2}$
8/10	0.069958	0.059820	0.049933	$6.0415 \cdot 10^{-2}$	$3.0023 \cdot 10^{-1}$	$1.3373 \cdot 10^{-1}$
9/10	0.069709	0.059334	0.049683	$4.1497 \cdot 10^{-1}$	1.1107	$6.3323 \cdot 10^{-1}$
10/7	0.070349	0.059886	0.050589	$-4.9922 \cdot 10^{-1}$	$2.0671 \cdot 10^{-1}$	-1.1788
10/8	0.070014	0.058526	0.049267	$-2.0051 \cdot 10^{-2}$	2.4568	1.4664
10/9	0.070130	0.060172	0.049735	$-1.8638 \cdot 10^{-1}$	$-2.8655 \cdot 10^{-1}$	$5.3002 \cdot 10^{-1}$
10/10	0.069253	0.060329	0.050747	1.0669	$-4.9657 \cdot 10^{-1}$	-1.4938
10/11	0.070132	0.060293	0.050262	$-1.8860 \cdot 10^{-1}$	$-4.8775 \cdot 10^{-1}$	$-5.2342 \cdot 10^{-1}$
10/12	0.069867	0.059831	0.049491	$1.8976 \cdot 10^{-1}$	$2.8188 \cdot 10^{-1}$	1.0189
10/13	0.070121	0.060359	0.049956	$-1.7412 \cdot 10^{-1}$	$-5.9849 \cdot 10^{-1}$	$8.8215 \cdot 10^{-2}$
10/14	0.069635	0.060407	0.051301	$5.2045 \cdot 10^{-1}$	$-6.7877 \cdot 10^{-1}$	-2.6019
10/15	0.070140	0.060561	0.050198	$-2.0043 \cdot 10^{-1}$	$-9.3507 \cdot 10^{-1}$	$-3.9554 \cdot 10^{-1}$
10/20	0.069928	0.059577	0.049514	$1.0228 \cdot 10^{-1}$	$7.0449 \cdot 10^{-1}$	$9.7144 \cdot 10^{-1}$
11/10	0.070085	0.059725	0.049792	$-1.2124 \cdot 10^{-1}$	$4.5791 \cdot 10^{-1}$	$4.1659 \cdot 10^{-1}$
12/10	0.070405	0.060919	0.050290	$-5.7814 \cdot 10^{-1}$	-1.5332	$-5.8024 \cdot 10^{-1}$
13/10	0.070318	0.060442	0.049915	$-4.5451 \cdot 10^{-1}$	$-7.3720 \cdot 10^{-1}$	$1.7019 \cdot 10^{-1}$
14/10	0.070721	0.061543	0.050771	-1.0301	-2.5708	-1.5422
15/10	0.070098	0.059889	0.049673	$-1.4089 \cdot 10^{-1}$	$1.8543 \cdot 10^{-1}$	$6.5406 \cdot 10^{-1}$
7/10/10	0.069543	0.058546	0.049302	$6.5351 \cdot 10^{-1}$	2.4239	1.3954
8/10/10	0.070369	0.059489	0.050213	$-5.2686 \cdot 10^{-1}$	$8.5068 \cdot 10^{-1}$	$-4.2509 \cdot 10^{-1}$
9/10/10	0.070006	0.060004	0.049999	$-8.6202 \cdot 10^{-3}$	$-6.3179 \cdot 10^{-3}$	$1.6787 \cdot 10^{-3}$
10/7/10	0.069980	0.060030	0.050059	$2.8367 \cdot 10^{-2}$	$-5.0014 \cdot 10^{-2}$	$-1.1829 \cdot 10^{-1}$
10/8/10	0.070259	0.060117	0.049731	$-3.6968 \cdot 10^{-1}$	$-1.9428 \cdot 10^{-1}$	$5.3877 \cdot 10^{-1}$
10/9/10	0.069455	0.061069	0.050188	$7.7848 \cdot 10^{-1}$	-1.7824	$-3.7527 \cdot 10^{-1}$
10/10/7	0.070029	0.060006	0.049961	$-4.1617 \cdot 10^{-2}$	$-9.3546 \cdot 10^{-3}$	$7.7302 \cdot 10^{-1}$
10/10/8	0.070041	0.060133	0.049880	$-5.8419 \cdot 10^{-2}$	$-2.2234 \cdot 10^{-1}$	$2.4010 \cdot 10^{-1}$
10/10/9	0.070292	0.059517	0.049832	$-4.1653 \cdot 10^{-1}$	$8.0422 \cdot 10^{-1}$	$3.3691 \cdot 10^{-1}$
10/10/10	0.070014	0.059989	0.049995	$-2.0333 \cdot 10^{-2}$	$1.7479 \cdot 10^{-2}$	$9.6699 \cdot 10^{-3}$
10/10/11	0.069868	0.060595	0.050485	$1.8804 \cdot 10^{-1}$	$-9.9089 \cdot 10^{-1}$	$-9.7055 \cdot 10^{-1}$
10/10/12	0.070218	0.061001	0.049996	$-3.1117 \cdot 10^{-1}$	-1.6679	$7.8766 \cdot 10^{-3}$
10/10/13	0.069266	0.059652	0.050306	1.0479	$5.8007 \cdot 10^{-1}$	$-6.1152 \cdot 10^{-1}$
10/10/14	0.069835	0.060840	0.051299	$2.3581 \cdot 10^{-1}$	-1.4008	-2.5999
10/10/15	0.070071	0.059569	0.049938	$-1.0209 \cdot 10^{-1}$	$7.1816 \cdot 10^{-1}$	$1.2310 \cdot 10^{-1}$
10/10/20	0.069980	0.060175	0.049346	$2.8397 \cdot 10^{-2}$	$-2.9185 \cdot 10^{-1}$	1.3075
10/11/10	0.069769	0.059090	0.049844	$3.3069 \cdot 10^{-1}$	1.5160	$3.1209 \cdot 10^{-1}$
10/12/10	0.070152	0.059770	0.050148	$-2.1662 \cdot 10^{-1}$	$3.8327 \cdot 10^{-1}$	$-2.9648 \cdot 10^{-1}$
10/13/10	0.069671	0.059522	0.050613	$4.6948 \cdot 10^{-1}$	$7.9699 \cdot 10^{-1}$	-1.2260
10/14/10	0.069732	0.060658	0.050709	$3.8243 \cdot 10^{-1}$	-1.0970	-1.4191
10/15/10	0.069938	0.060644	0.049861	$8.7938 \cdot 10^{-2}$	-1.0746	$2.7704 \cdot 10^{-1}$
10/20/10	0.069469	0.061147	0.050542	$7.5834 \cdot 10^{-1}$	-1.9109	-1.0835
11/10/10	0.069865	0.060484	0.050890	$1.9282 \cdot 10^{-1}$	$-8.0663 \cdot 10^{-1}$	-1.7809
12/10/10	0.069517	0.060659	0.050161	$6.9018 \cdot 10^{-1}$	-1.0980	$-3.2236 \cdot 10^{-1}$
13/10/10	0.069458	0.062045	0.050784	$7.7388 \cdot 10^{-1}$	-3.4089	-1.5682
14/10/10	0.070055	0.060206	0.049897	$-7.8010 \cdot 10^{-2}$	$-3.4310 \cdot 10^{-1}$	$2.0571 \cdot 10^{-1}$
15/10/10	0.069970	0.059599	0.049704	$4.2373 \cdot 10^{-2}$	$6.6829 \cdot 10^{-1}$	$5.9187 \cdot 10^{-1}$
20/10/10	0.069528	0.059828	0.050387	$6.7434 \cdot 10^{-1}$	$2.8719 \cdot 10^{-1}$	$-7.7465 \cdot 10^{-1}$

Tabla VII.4.1: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando k_{12} , k_{21} y k_{23} son computados del modelo **D**.

Una vez realizado satisfactoriamente este proceso, generamos para todos y cada uno de los 3 modelos, 8 curvas cinéticas, con 50 valores de absorbancia cada uno y ternas de valores de constantes cinéticas diferentes a la utilizadas para generar los datos sintéticos con los que se ha llevado a cabo el *Entrenamiento*. A los datos cinéticos se les impone un valor de *noise* ($\pm 1.10^{-4}$)

y se distribuyen en 3 matrices de *Inputs*, con dimensiones 8x50, correspondientes a cada uno de los tres modelos. Estas matrices son para realizar el proceso de *Predicción* de la siguiente forma. Suministramos a la Red Neuronal ya entrenada exclusivamente para el modelo *I*, las 3 matrices de *Inputs* de cada uno de los 3 modelos proporcionándonos en cada caso las correspondientes matrices de *Outputs [Y]* (dimensiones 59x3), que contienen los valores de las constantes cinéticas que la Red Neuronal *Predice* para cada modelo (Tablas VII.4.2, VII.4.3 y VII.4.4).

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
0.058750	0.048750	0.061250	0.057558	0.056723	0.068536	2.0398	-16.355	-11.895
0.066250	0.056250	0.053750	0.066254	0.056252	0.053750	$-5.7021 \cdot 10^{-3}$	$-3.6202 \cdot 10^{-3}$	$-6.9173 \cdot 10^{-4}$
0.073750	0.063750	0.046250	0.073752	0.063751	0.046243	$-2.6278 \cdot 10^{-3}$	$-1.9126 \cdot 10^{-3}$	$1.5881 \cdot 10^{-2}$
0.081250	0.071250	0.038750	0.081336	0.071349	0.038739	$-1.0617 \cdot 10^{-1}$	$-1.3852 \cdot 10^{-1}$	$2.6143 \cdot 10^{-2}$
0.081250	0.071250	0.061250	0.081259	0.071102	0.061205	$-1.0822 \cdot 10^{-2}$	$2.0741 \cdot 10^{-1}$	$7.3227 \cdot 10^{-2}$
0.073750	0.063750	0.053750	0.073750	0.063775	0.053756	$-2.7750 \cdot 10^{-4}$	$-3.9645 \cdot 10^{-2}$	$-1.1588 \cdot 10^{-2}$
0.066250	0.056250	0.046250	0.066252	0.056249	0.046253	$-3.5211 \cdot 10^{-3}$	$-8.6538 \cdot 10^{-3}$	$-7.1884 \cdot 10^{-3}$
0.058750	0.048750	0.038750	0.058739	0.048704	0.038729	$1.8037 \cdot 10^{-2}$	$9.5342 \cdot 10^{-2}$	$5.2121 \cdot 10^{-2}$

Tabla VII.4.2: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{21} y k_{23} cuando ha sido aplicada la Red con una Arquitectura óptima (50, 3, 3) y Configuración (9/10/10) en el modelo *I*.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
0.058750	0.048750	0.061250	0.089427	-0.15925	-0.073279	-52.217	426.77	219.64
0.066250	0.056250	0.053750	0.10024	-0.16775	-0.072526	-51.304	398.22	234.93
0.073750	0.063750	0.046250	0.10707	-0.17533	-0.071168	-45.183	375.03	253.88
0.081250	0.071250	0.038750	0.11084	-0.18149	-0.069773	-36.421	354.72	280.06
0.081250	0.071250	0.061250	0.11688	-0.17155	-0.060305	-43.855	340.76	198.46
0.073750	0.063750	0.053750	0.11065	-0.17813	-0.067629	-50.030	379.41	225.82
0.066250	0.056250	0.046250	0.098805	-0.17031	-0.069111	-49.139	402.76	249.43
0.058750	0.048750	0.038750	0.086149	-0.16612	-0.065426	-46.636	440.76	268.84

Tabla VII.4.3: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{23} y k_{32} cuando ha sido aplicada la Red con una Arquitectura óptima (50, 3, 3) y Configuración (9/10/10) en el modelo *II*.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
0.058750	0.048750	0.061250	0.093433	-0.082678	-0.027856	-59.034	269.60	145.58
0.066250	0.056250	0.053750	0.10915	-0.085810	-0.025553	-64.753	252.55	147.54
0.073750	0.063750	0.046250	0.11904	-0.086761	-0.023484	-61.415	236.10	150.78
0.081250	0.071250	0.038750	0.12469	-0.093002	-0.024759	-53.474	230.53	163.89
0.081250	0.071250	0.061250	0.12219	-0.060237	-0.0091367	-50.395	184.54	114.92
0.073750	0.063750	0.053750	0.11869	-0.074851	-0.017442	-60.930	217.41	132.45
0.066250	0.056250	0.046250	0.10853	-0.10044	-0.033397	-63.824	278.56	172.21
0.058750	0.048750	0.038750	0.090642	-0.13174	-0.055338	-54.284	370.25	242.81

Tabla VII.4.4: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{23} y k_{31} cuando ha sido aplicada la Red con una Arquitectura óptima (50,3,3) y Configuración (9/10/10) en el modelo **III**.

Analizando estas 3 matrices de *Outputs* [Y], observamos que exclusivamente en la que corresponde al proceso de *Predicción* del modelo base (**I**), existe una coincidencia entre los valores de las constantes cinéticas de *outputs* con los que sirvieron para generar las 8 curvas. Sin embargo, en las otras 2 matrices [Y] correspondientes a los modelos **II**) y **III**), no solo es que no haya concordancia, sino que se obtienen valores absurdos (incluso valores negativos de las k cinéticas) que ponen de manifiesto el fracaso en la *Predicción* de los valores de constantes cinéticas para estos 2 modelos con la Red ANN *entrenada* exclusivamente para el caso **aI**). A la vista de los resultados obtenidos, podemos concluir que la Red Neuronal (ANN) es capaz de distinguir entre modelos diferentes en el proceso de *Predicción* conduciendo a valores correctos de constantes de velocidad únicamente cuando se le suministran los datos cinéticos de *Input* pertenecientes al modelo que sirvió para su *Entrenamiento*. En caso contrario, cuando no pertenece a dicho modelo, el proceso de *Predicción* fracasa, obteniendo valores absurdos y por tanto, inaceptables.

VII.4.1.2.2- Modelo II

Procediendo de igual manera que en el caso anterior, elegimos “*a priori*” como base el modelo **II**). A partir del correspondiente ED (Figura VII.4.1) hemos generado una serie de curvas cinéticas para este mecanismo con las que se lleva a cabo el proceso de *Entrenamiento* al objeto de determinar la Arquitectura óptima para realizar el proceso de *Predicción* de las constantes de velocidad. La Tabla VII.4.5 muestra los resultados obtenidos en el *Entrenamiento* de la ANN con diferentes Estructuras y Configuraciones.

Config.	ANN		Artificial	Neural	Networks	
	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
7/10	0.070001	0.059995	0.050000	$-9.2447 \cdot 10^{-4}$	$8.3888 \cdot 10^{-3}$	$-2.9054 \cdot 10^{-5}$
8/10	0.070001	0.059999	0.050000	$-1.1286 \cdot 10^{-3}$	$3.0303 \cdot 10^{-4}$	$-5.2126 \cdot 10^{-4}$
9/10	0.069997	0.059998	0.050002	$4.8133 \cdot 10^{-3}$	$2.5504 \cdot 10^{-3}$	$-3.8371 \cdot 10^{-3}$
10/7	0.069993	0.059979	0.050022	$1.0212 \cdot 10^{-2}$	$3.3893 \cdot 10^{-2}$	$-4.4062 \cdot 10^{-2}$
10/8	0.069999	0.059999	0.049999	$1.0108 \cdot 10^{-3}$	$7.1469 \cdot 10^{-4}$	$8.8640 \cdot 10^{-4}$
10/9	0.069999	0.059999	0.049999	$2.1579 \cdot 10^{-4}$	$8.3560 \cdot 10^{-4}$	$1.9713 \cdot 10^{-3}$
10/10	0.070000	0.059999	0.050001	$-4.6239 \cdot 10^{-5}$	$7.4591 \cdot 10^{-4}$	$-1.3239 \cdot 10^{-3}$
10/11	0.070000	0.060001	0.049999	$-3.9533 \cdot 10^{-4}$	$-9.1897 \cdot 10^{-4}$	$2.0263 \cdot 10^{-3}$
10/12	0.069999	0.059998	0.049998	$4.2570 \cdot 10^{-4}$	$4.0628 \cdot 10^{-4}$	$3.2378 \cdot 10^{-3}$
10/13	0.069999	0.059999	0.049999	$9.5338 \cdot 10^{-5}$	$3.6528 \cdot 10^{-4}$	$2.2151 \cdot 10^{-3}$
10/14	0.069999	0.059999	0.049999	$1.3177 \cdot 10^{-4}$	$9.2303 \cdot 10^{-4}$	$1.9183 \cdot 10^{-3}$
10/15	0.069996	0.059959	0.049997	$5.6749 \cdot 10^{-3}$	$6.8143 \cdot 10^{-2}$	$4.9816 \cdot 10^{-3}$
10/20	0.069999	0.059999	0.049999	$1.3741 \cdot 10^{-4}$	$4.0092 \cdot 10^{-4}$	$1.4902 \cdot 10^{-4}$
11/10	0.069991	0.059997	0.050001	$1.2307 \cdot 10^{-4}$	$5.6525 \cdot 10^{-3}$	$-1.0356 \cdot 10^{-3}$
12/10	0.069995	0.059996	0.050000	$7.8448 \cdot 10^{-3}$	$7.4983 \cdot 10^{-3}$	$-3.3107 \cdot 10^{-4}$
13/10	0.070000	0.060000	0.049997	$-2.0401 \cdot 10^{-4}$	$-1.1478 \cdot 10^{-5}$	$6.8041 \cdot 10^{-5}$
14/10	0.070001	0.059995	0.049997	$-1.1138 \cdot 10^{-3}$	$8.8918 \cdot 10^{-3}$	$6.1104 \cdot 10^{-3}$
15/10	0.070001	0.059999	0.049996	$-1.0251 \cdot 10^{-3}$	$5.8405 \cdot 10^{-4}$	$5.9343 \cdot 10^{-3}$
20/10	0.069992	0.060004	0.049996	$1.2097 \cdot 10^{-2}$	$-6.5885 \cdot 10^{-3}$	$8.4103 \cdot 10^{-3}$
7/10/10	0.070002	0.059999	0.050000	$-2.8321 \cdot 10^{-3}$	$1.8349 \cdot 10^{-3}$	$-8.8200 \cdot 10^{-4}$
8/10/10	0.069999	0.059999	0.049999	$4.6044 \cdot 10^{-4}$	$5.0700 \cdot 10^{-4}$	$1.5666 \cdot 10^{-3}$
9/10/10	0.069999	0.059999	0.049998	$6.7924 \cdot 10^{-4}$	$2.0886 \cdot 10^{-3}$	$3.5461 \cdot 10^{-3}$
10/7/10	0.070000	0.060001	0.050000	$-7.0243 \cdot 10^{-4}$	$-1.4400 \cdot 10^{-3}$	$-4.9772 \cdot 10^{-4}$
10/8/10	0.070000	0.059999	0.050000	$-3.1046 \cdot 10^{-4}$	$2.1018 \cdot 10^{-4}$	$-8.4650 \cdot 10^{-4}$
10/9/10	0.070000	0.060000	0.049999	$-1.5085 \cdot 10^{-4}$	$-2.0798 \cdot 10^{-4}$	$5.8371 \cdot 10^{-4}$
10/10/7	0.069998	0.059999	0.050001	$3.1671 \cdot 10^{-3}$	$1.0228 \cdot 10^{-3}$	$-2.5088 \cdot 10^{-3}$
10/10/8	0.069999	0.059999	0.050001	$8.9695 \cdot 10^{-3}$	$1.0302 \cdot 10^{-3}$	$-2.4542 \cdot 10^{-3}$
10/10/9	0.069999	0.060000	0.050001	$1.6819 \cdot 10^{-4}$	$-6.2954 \cdot 10^{-4}$	$-2.5331 \cdot 10^{-3}$
10/10/10	0.070000	0.060000	0.049999	$-3.0785 \cdot 10^{-4}$	$-2.1263 \cdot 10^{-4}$	$1.8421 \cdot 10^{-4}$
10/10/11	0.069989	0.059993	0.050009	$1.4531 \cdot 10^{-2}$	$1.1635 \cdot 10^{-2}$	$-1.7248 \cdot 10^{-2}$
10/10/12	0.069999	0.060000	0.049999	$2.0637 \cdot 10^{-4}$	$-2.2155 \cdot 10^{-4}$	$2.9828 \cdot 10^{-4}$
10/10/13	0.070000	0.060000	0.050000	$-9.3736 \cdot 10^{-4}$	$-2.9780 \cdot 10^{-4}$	$-3.0704 \cdot 10^{-5}$
10/10/14	0.070000	0.059999	0.049999	$-1.4238 \cdot 10^{-4}$	$4.5565 \cdot 10^{-5}$	$8.4049 \cdot 10^{-4}$
10/10/15	0.069999	0.059998	0.049999	$5.3319 \cdot 10^{-4}$	$3.2439 \cdot 10^{-3}$	$6.8800 \cdot 10^{-4}$
10/10/20	0.069999	0.059999	0.049999	$2.0124 \cdot 10^{-4}$	$7.5349 \cdot 10^{-4}$	$9.2039 \cdot 10^{-4}$
10/11/10	0.070000	0.059999	0.050000	$-3.9555 \cdot 10^{-4}$	$3.5388 \cdot 10^{-5}$	$-1.8423 \cdot 10^{-4}$
10/12/10	0.069999	0.060000	0.049997	$1.2542 \cdot 10^{-3}$	$-6.3472 \cdot 10^{-4}$	$6.9075 \cdot 10^{-3}$
10/13/10	0.069998	0.059990	0.049997	$2.2132 \cdot 10^{-3}$	$1.6272 \cdot 10^{-2}$	$6.1190 \cdot 10^{-3}$
10/14/10	0.069998	0.059997	0.049996	$2.2291 \cdot 10^{-3}$	$4.0680 \cdot 10^{-3}$	$7.5060 \cdot 10^{-3}$
10/15/10	0.070092	0.060044	0.049959	$-1.3127 \cdot 10^{-1}$	$-7.2702 \cdot 10^{-2}$	$8.2149 \cdot 10^{-3}$
10/20/10	0.069999	0.059998	0.049999	$2.8597 \cdot 10^{-5}$	$3.3380 \cdot 10^{-3}$	$9.8557 \cdot 10^{-4}$
11/10/10	0.069998	0.059997	0.049997	$1.6533 \cdot 10^{-3}$	$4.7184 \cdot 10^{-3}$	$6.2549 \cdot 10^{-3}$
12/10/10	0.070000	0.060000	0.049999	$-1.8286 \cdot 10^{-4}$	$-4.0747 \cdot 10^{-4}$	$2.1888 \cdot 10^{-4}$
13/10/10	0.070002	0.059997	0.049999	$-2.3777 \cdot 10^{-3}$	$5.7556 \cdot 10^{-3}$	$1.3633 \cdot 10^{-3}$
14/10/10	0.070053	0.060143	0.049809	$-7.5397 \cdot 10^{-3}$	$-2.3795 \cdot 10^{-1}$	$3.8135 \cdot 10^{-1}$
15/10/10	0.069996	0.059959	0.050043	$5.8667 \cdot 10^{-3}$	$6.7155 \cdot 10^{-2}$	$-8.5476 \cdot 10^{-2}$
20/10/10	0.069994	0.060000	0.049989	$9.0307 \cdot 10^{-3}$	$-1.5412 \cdot 10^{-4}$	$2.0704 \cdot 10^{-2}$

Tabla VII.4.5: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando k_{12} , k_{23} y k_{32} son computados del modelo **a2**).

Una vez realizado satisfactoriamente este proceso de *Entrenamiento* y conseguida la Arquitectura óptima, generamos para todos y cada uno de los 3 modelos, 8 curvas cinéticas, con 50 valores de absorbancia cada uno y ternas de valores de constantes cinéticas diferentes a la

utilizadas para generar las matrices *Inputs* con las que se lleva a cabo el *Entrenamiento* de la misma forma que en el caso anterior. Se ha impuesto un error de ($\pm 1.10^{-4}$) a los datos y se agrupan en 3 matrices de *Inputs*, con dimensiones 8x50. Cada una de estas matrices es sometida a un proceso de *Predicción* con la Red Neuronal *entrenada* exclusivamente para el modelo **II**, proporcionando en cada caso la matriz de *Outputs* [*Y*] (dimensiones 59x3), que contienen los valores de las constantes cinéticas que la Red Neuronal *Predice* para cada modelo. Los resultados de los procesos de *Predicción* realizados con la ANN entrenada para el modelo **II** se muestran en las Tablas VII.4.6, VII.4.7 y VII.4.8.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
0.058750	0.048750	0.061250	0.058750	0.048746	0.061248	$-5.7661 \cdot 10^{-4}$	$7.5270 \cdot 10^{-3}$	$2.9185 \cdot 10^{-3}$
0.066250	0.056250	0.053750	0.066250	0.056249	0.053750	$-7.3825 \cdot 10^{-5}$	$5.3688 \cdot 10^{-4}$	$-8.7737 \cdot 10^{-4}$
0.073750	0.063750	0.046250	0.073749	0.063751	0.046251	$3.3055 \cdot 10^{-4}$	$-2.1329 \cdot 10^{-3}$	$-1.7196 \cdot 10^{-3}$
0.081250	0.071250	0.038750	0.081250	0.071255	0.038749	$-2.0770 \cdot 10^{-4}$	$-7.0052 \cdot 10^{-4}$	$1.4365 \cdot 10^{-3}$
0.081250	0.071250	0.061250	0.081249	0.071253	0.061255	$1.4159 \cdot 10^{-3}$	$-3.5546 \cdot 10^{-3}$	$-7.5865 \cdot 10^{-3}$
0.073750	0.063750	0.053750	0.073750	0.063750	0.053749	$-3.3204 \cdot 10^{-4}$	$-5.6038 \cdot 10^{-4}$	$9.5832 \cdot 10^{-4}$
0.066250	0.056250	0.046250	0.066249	0.056254	0.046245	$4.0244 \cdot 10^{-4}$	$-6.8206 \cdot 10^{-4}$	$1.0641 \cdot 10^{-2}$
0.058750	0.048750	0.038750	0.058748	0.048727	0.038748	$3.0051 \cdot 10^{-3}$	$4.6896 \cdot 10^{-2}$	$5.8289 \cdot 10^{-3}$

Tabla VII.4.6: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{21} y k_{23} cuando ha sido aplicada la Red con una Arquitectura óptima (50, 3, 3) y Configuración (10/10/13) en el modelo **II**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
0.058750	0.048750	0.061250	0.058532	0.030347	0.0036363	0.37049	37.750	94.063
0.066250	0.056250	0.053750	0.063845	0.018147	-0.0017041	3.6309	67.739	103.17
0.073750	0.063750	0.046250	0.067102	0.0067477	-0.0092028	9.0139	89.415	119.90
0.081250	0.071250	0.038750	0.068089	-0.0042771	-0.018818	16.198	106.00	148.56
0.081250	0.071250	0.061250	0.074876	0.018342	-0.0013348	7.8454	74.257	102.18
0.073750	0.063750	0.053750	0.068746	0.014279	-0.0039868	6.7857	77.602	107.42
0.066250	0.056250	0.046250	0.062875	0.010582	-0.0062787	5.0950	81.188	113.58
0.058750	0.048750	0.038750	0.057543	0.0073420	-0.0081421	2.0537	84.939	121.01

Tabla VII.4.7: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{23} y k_{32} cuando ha sido aplicada la Red con una Arquitectura óptima (50,3,3) y Configuración (10/10/13) en el modelo **I**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
0.058750	0.048750	0.061250	0.049615	0.081397	0.20273	15.549	-66.968	-230.99
0.066250	0.056250	0.053750	0.054558	0.088577	0.20041	17.648	-57.470	-272.86
0.073750	0.063750	0.046250	0.060618	0.094868	0.19206	17.806	-48.813	-315.26
0.081250	0.071250	0.038750	0.068455	0.097727	0.17023	15.748	-37.161	-339.30
0.081250	0.071250	0.061250	0.065028	0.10060	0.20697	19.966	-41.197	-237.91
0.073750	0.063750	0.053750	0.059957	0.095843	0.20242	18.702	-50.342	-276.59
0.066250	0.056250	0.046250	0.055063	0.085858	0.18869	16.886	-52.636	-307.99
0.058750	0.048750	0.038750	0.051452	0.065286	0.15579	12.421	-33.920	-302.03

Tabla VII.4.8: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{23} y k_{31} cuando ha sido aplicada la Red con una Arquitectura óptima (50,3,3) y Configuración (10/10/13) en el modelo **III**.

Al igual que en el caso anterior, únicamente la matriz $[Y]$ que corresponde al proceso de *Predicción* del modelo tomado como base (**II**) en este caso proporciona los valores de las constantes cinéticas con los que han generado las 8 curvas.

VII.4.1.2.3- Modelo III

Tal y como hemos hecho en los casos anteriores, hemos elegido *a priori* como modelo base el **III**). Hemos generado una serie de curvas cinéticas para este mecanismo con las que se ha llevado a cabo el proceso de *Entrenamiento* considerando diferentes Estructuras y Configuraciones y se ha determinado la Arquitectura óptima para realizar el proceso de *Predicción* de las constantes de velocidad. La Tabla VII.4.9 muestra los resultados obtenidos en el *Entrenamiento* de la ANN.

ANN		<i>Artificial</i>		<i>Neural Networks</i>		
Config.	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
7/10	0.070257	0.060378	0.049664	-3.6745.10 ⁻¹	-6.2944.10 ⁻¹	6.7124.10 ⁻¹
8/10	0.070179	0.060311	0.049906	-2.5599.10 ⁻¹	-5.1852.10 ⁻¹	1.8820.10 ⁻¹
9/10	0.069976	0.060010	0.050044	3.4678.10 ⁻²	-1.6959.10 ⁻²	-8.8802.10 ⁻²
10/7	0.070291	0.060296	0.049717	-4.1632.10 ⁻¹	-4.9273.10 ⁻¹	5.6664.10 ⁻¹
10/8	0.069899	0.060832	0.049387	1.4439.10 ⁻¹	-1.3862	1.2266
10/9	0.070422	0.059884	0.049949	-6.0234.10 ⁻¹	1.9326.10 ⁻¹	1.0024.10 ⁻¹
10/10	0.070119	0.060261	0.050062	-1.6984.10 ⁻¹	-4.3527.10 ⁻¹	-1.2376.10 ⁻¹
10/11	0.069850	0.060245	0.050098	2.1427.10 ⁻¹	-4.0879.10 ⁻¹	-1.9640.10 ⁻¹
10/12	0.070047	0.059949	0.049975	-6.6846.10 ⁻²	8.4701.10 ⁻²	4.9423.10 ⁻²
10/13	0.069943	0.060043	0.049893	8.0717.10 ⁻²	-7.2361.10 ⁻²	2.2132.10 ⁻¹
10/14	0.069922	0.060047	0.050101	1.1136.10 ⁻¹	-7.8313.10 ⁻²	-2.0261.10 ⁻¹
10/15	0.070003	0.059999	0.050001	-4.4877.10 ⁻³	2.1908.10 ⁻³	-2.4284.10 ⁻³
10/20	0.069979	0.059927	0.050142	3.0578.10 ⁻²	1.2208.10 ⁻¹	-2.8442.10 ⁻¹
11/10	0.070067	0.059920	0.049944	-9.6047.10 ⁻²	1.3339.10 ⁻¹	1.1262.10 ⁻¹
12/10	0.069776	0.060388	0.049829	3.1971.10 ⁻¹	-6.4609.10 ⁻¹	3.4023.10 ⁻¹
13/10	0.070042	0.060014	0.049842	-5.9288.10 ⁻²	-2.3287.10 ⁻²	3.1658.10 ⁻¹
14/10	0.069893	0.059971	0.050001	1.5354.10 ⁻¹	4.8443.10 ⁻²	-2.9788.10 ⁻³
15/10	0.070058	0.060300	0.050107	-8.3016.10 ⁻²	-5.0033.10 ⁻¹	-2.1465.10 ⁻¹
20/10	0.069833	0.060136	0.049956	2.3870.10 ⁻¹	-2.2625.10 ⁻¹	8.7168.10 ⁻²
7/10/10	0.069967	0.059814	0.050061	4.6496.10 ⁻²	3.0935.10 ⁻¹	-1.2272.10 ⁻¹
8/10/10	0.070032	0.060416	0.050034	-4.5488.10 ⁻²	-6.9295.10 ⁻¹	-6.7526.10 ⁻²
9/10/10	0.069992	0.059995	0.050041	1.0909.10 ⁻²	8.3268.10 ⁻³	-8.1490.10 ⁻²
10/7/10	0.070113	0.059921	0.050105	-1.6092.10 ⁻¹	1.3193.10 ⁻¹	-2.1056.10 ⁻¹
10/8/10	0.070049	0.060148	0.049974	-7.0145.10 ⁻²	-2.4625.10 ⁻¹	5.2049.10 ⁻²
10/9/10	0.069986	0.059983	0.050001	1.9896.10 ⁻²	2.7027.10 ⁻²	-2.6050.10 ⁻³
10/10/7	0.069880	0.060981	0.049400	1.7127.10 ⁻¹	-1.6349	1.1990
10/10/8	0.069978	0.059981	0.050009	3.1602.10 ⁻²	3.1607.10 ⁻²	-1.7417.10 ⁻²
10/10/9	0.069802	0.061084	0.049929	2.8230.10 ⁻¹	-1.8071	1.4123.10 ⁻¹
10/10/10	0.070080	0.060555	0.049994	-1.1487.10 ⁻¹	-9.2542.10 ⁻¹	1.1796.10 ⁻²
10/10/11	0.070169	0.059849	0.049821	-2.4080.10 ⁻¹	2.5018.10 ⁻¹	3.5759.10 ⁻¹
10/10/12	0.070341	0.060778	0.050270	-4.8762.10 ⁻¹	-1.2967	-5.4024.10 ⁻¹
10/10/13	0.070000	0.060000	0.049999	-1.0884.10⁻⁴	-7.9073.10⁻⁵	8.8021.10⁻⁵
10/10/14	0.069810	0.060244	0.050113	2.7079.10 ⁻¹	-4.0668.10 ⁻¹	-2.2522.10 ⁻¹
10/10/15	0.069977	0.060425	0.050108	3.3539.10 ⁻²	-7.0823.10 ⁻¹	-2.1589.10 ⁻¹
10/10/20	0.070235	0.059789	0.050692	-3.3566.10 ⁻¹	3.5000.10 ⁻¹	-1.3837
10/11/10	0.069829	0.060203	0.049957	2.4390.10 ⁻¹	-3.3785.10 ⁻¹	8.6274.10 ⁻²
10/12/10	0.070175	0.060232	0.050309	-2.5039.10 ⁻¹	-3.8674.10 ⁻¹	-6.1918.10 ⁻¹
10/13/10	0.069859	0.060448	0.049898	2.0009.10 ⁻¹	-7.4678.10 ⁻¹	2.0315.10 ⁻¹
10/14/10	0.070001	0.060127	0.049709	-1.6613.10 ⁻³	-2.1088.10 ⁻¹	5.8066.10 ⁻¹
10/15/10	0.070001	0.060000	0.049999	-8.6092.10 ⁻⁴	-2.6059.10 ⁻⁴	1.0269.10 ⁻³
10/20/10	0.069703	0.060977	0.050053	4.2399.10 ⁻¹	-1.6277	-1.0662.10 ⁻¹
11/10/10	0.069897	0.060133	0.049673	1.4701.10 ⁻¹	-2.2171.10 ⁻¹	6.5448.10 ⁻¹
12/10/10	0.069929	0.060516	0.050065	1.0050.10 ⁻¹	-8.6033.10 ⁻¹	-1.3079.10 ⁻¹
13/10/10	0.069983	0.060563	0.049564	2.4195.10 ⁻²	-9.3766.10 ⁻¹	8.7139.10 ⁻¹
14/10/10	0.070015	0.060041	0.049798	-2.0961.10 ⁻²	-6.7972.10 ⁻²	4.0416.10 ⁻¹
15/10/10	0.070261	0.060370	0.049769	-3.7217.10 ⁻¹	-6.1690.10 ⁻¹	4.6177.10 ⁻¹
20/10/10	0.069861	0.060268	0.049562	1.9793.10 ⁻¹	-4.2972.10 ⁻¹	8.7620.10 ⁻¹

Tabla VII.4.9: Resultados del proceso de *Entrenamiento* de la Red Neuronal constituida por una Arquitectura con 2 y 3 *hidden layers* y distintas Configuraciones, cuando k_{12} , k_{23} y k_{31} son computados del modelo **a3**).

Posteriormente generamos para cada uno de los 3 sistemas, 8 curvas cinéticas, con 50 valores de absorbancia a partir de ternas de valores de constantes cinéticas diferentes a la utilizadas para generar los datos sintéticos para llevar a cabo el *Entrenamiento* de la misma

manera que en los casos anteriores. Se han construido 3 matrices de *Inputs*, con dimensiones 8x50, cada una de estas matrices es sometida a un proceso de *Predicción* con la Red Neuronal ya *entrenada* exclusivamente para el modelo **III**, proporcionando en cada caso la matriz de *Outputs* [*Y*] (dimensiones 59x3), que contienen los valores de las constantes cinéticas que la Red Neuronal *Predice*, para cada modelo correspondientes a cada uno de los al procesos de *Predicción* individual (Tablas VII.4.10, VII.4.11 y VII.4.12).

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
0.058750	0.048750	0.061450	0.058749	0.048749	0.061246	$7.9226 \cdot 10^{-4}$	$7.2854 \cdot 10^{-4}$	$6.3928 \cdot 10^{-3}$
0.066250	0.056250	0.053750	0.066250	0.056249	0.053751	$-2.0241 \cdot 10^{-1}$	$3.9991 \cdot 10^{-5}$	$-1.2650 \cdot 10^{-3}$
0.073750	0.063750	0.046250	0.073750	0.063749	0.046249	$-7.6214 \cdot 10^{-5}$	$2.0942 \cdot 10^{-4}$	$3.5496 \cdot 10^{-4}$
0.081250	0.071250	0.038750	0.081255	0.071247	0.038732	$-5.9079 \cdot 10^{-3}$	$4.1696 \cdot 10^{-3}$	$4.7718 \cdot 10^{-2}$
0.081250	0.071250	0.061250	0.081252	0.071253	0.061252	$-2.3194 \cdot 10^{-3}$	$-5.3504 \cdot 10^{-3}$	$-3.9707 \cdot 10^{-3}$
0.073750	0.063750	0.053750	0.073750	0.063749	0.053750	$-2.7557 \cdot 10^{-4}$	$4.1710 \cdot 10^{-4}$	$-2.5992 \cdot 10^{-4}$
0.066250	0.056250	0.046250	0.066250	0.056250	0.046249	$-2.3049 \cdot 10^{-4}$	$-7.6780 \cdot 10^{-4}$	$8.1915 \cdot 10^{-4}$
0.058750	0.048750	0.038750	0.058749	0.048749	0.038752	$7.9246 \cdot 10^{-4}$	$2.0235 \cdot 10^{-3}$	$-4.5959 \cdot 10^{-3}$

Tabla VII.4.10: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{23} y k_{31} cuando ha sido aplicada la Red con una Arquitectura óptima (50, 3, 3) y Configuración (10/10/13) en el modelo **III**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
0.058750	0.048750	0.061450	0.058299	0.026428	-0.0038367	0.76696	45.789	106.26
0.066250	0.056250	0.053750	0.063379	0.016309	-0.012654	4.3329	71.007	123.54
0.073750	0.063750	0.046250	0.066422	0.0067369	-0.024765	9.9359	89.432	153.55
0.081250	0.071250	0.038750	0.068886	-0.0016629	-0.033571	15.218	102.33	186.64
0.081250	0.071250	0.061250	0.075308	0.015969	-0.018739	7.3135	77.588	130.55
0.073750	0.063750	0.053750	0.068439	0.012873	-0.019109	7.2003	79.807	135.55
0.066250	0.056250	0.046250	0.061808	0.010063	-0.017623	6.7049	82.110	138.10
0.058750	0.048750	0.038750	0.055131	0.0073121	-0.014082	6.1598	85.001	136.34

Tabla VII.4.11: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{21} y k_{23} cuando ha sido aplicada la Red con una Arquitectura óptima (50,3,3) y Configuración (10/10/13) en el modelo **I**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
0.058750	0.048750	0.061450	0.058552	0.046232	0.021497	0.33689	5.1649	64.902
0.066250	0.056250	0.053750	0.066059	0.053707	0.019136	0.28820	4.5215	64.399
0.073750	0.063750	0.046250	0.073704	0.061493	0.016664	0.062049	3.5402	63.969
0.081250	0.071250	0.038750	0.081586	0.069527	0.013964	-0.41298	2.4187	63.964
0.081250	0.071250	0.061250	0.081465	0.067412	0.020655	-0.26417	5.3866	66.278
0.073750	0.063750	0.053750	0.073696	0.060922	0.018846	0.073008	4.4365	64.937
0.066250	0.056250	0.046250	0.066059	0.054141	0.016959	0.28698	3.7498	63.332
0.058750	0.048750	0.038750	0.058642	0.047189	0.014808	0.18318	3.2000	61.786

Tabla VII.4.12: Resultados del proceso de *Predicción* obtenidos después de la aplicación ANN para k_{12} , k_{23} y k_{32} cuando ha sido aplicada la Red con una Arquitectura óptima (50,3,3) y Configuración (10/10/13) en el modelo **II**.

Tal y como puede comprobarse en las Tablas VII.4.10, VII.4.11 y VII.4.12, únicamente la matriz $[Y]$ que corresponde al proceso de *Predicción* del modelo base (**III**) proporciona los valores de las constantes cinéticas con los que han generado las 8 curvas.

VII.4.2.- Discriminación de mecanismos de reacción mediante AGDC

VII.4.2.1.- Determinación Constantes velocidad mediante la aplicación del algoritmo AGDC. Discriminación de mecanismos de reacción

El algoritmo de Optimización Matemática AGDC se ha empleado para la determinación de las constantes de velocidad de las etapas que constituyen determinados mecanismos o modelos de reacción. Los valores de las constantes que han sido obtenidos mediante el proceso de *Predicción* realizado con ANN, son sometidos a un proceso de optimización matemática mediante la aplicación computacional **KINMODEL(AGDC)@Vs.1.0** que va a permitir la discriminación de modelos y validación del que mejor se ajuste a los datos experimentales, obviamente los mismos ya estudiados mediante ANN [I, II) y III)] en la primera etapa del tratamiento completo que realiza el algoritmo híbrido AH (ANN-AGDC).

El procedimiento que hemos seguido para el estudio de la capacidad de discriminación entre diversos modelos o mecanismos de reacción mediante el algoritmo AGDC es idéntico al ya explicado en detalle para ANN. En este caso la aplicación computacional del algoritmo AGDC la hemos realizado mediante el programa **KINMODEL(AGDC)@Vs.1.0**.

VII.4.2.2.- Resultados de la discriminación mediante AGDC

Por último hemos aplicado el algoritmo de optimización matemática AGDC para la discriminación entre mecanismos o modelos de reacción mediante la determinación de las constantes de velocidad de las etapas que constituyen el mecanismo. Para realizar el proceso de

optimización hemos utilizado como estimaciones iniciales los valores de las constantes que han sido obtenidos mediante el proceso de *Predicción* realizado con ANN de acuerdo a las exigencias del algoritmo AH. El procedimiento es el siguiente:

1) Consideramos en primer lugar el mecanismo **I**):

- Generamos datos cinéticos de absorbancia/tiempo a partir de unos determinados valores de las constantes de velocidad.

- Suponemos desconocidos los valores de la constantes de velocidad, a partir de los datos cinéticos de absorbancia mediante el programa **KINMODEL(AGDC)** se determinan los valores de las mismas considerando los mecanismos de reacción **I**, **II** y **III**. Las Tablas VII.4.13, VII.4.14 y VII.4.15 muestran los resultados obtenidos en cada caso.

- El análisis de los resultados y los parámetros estadísticos obtenidos han permitido decidir cuál es el mecanismo que mejor representa los datos absorbancia. En este caso se corresponde con el mecanismo **I**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
0.057558	0.056723	0.068536	0.058727	0.049129	0.061546	$3.8701 \cdot 10^{-2}$	$-7.7703 \cdot 10^{-1}$	$-4.8398 \cdot 10^{-1}$
0.066254	0.056252	0.053750	0.066253	0.056267	0.053759	$-4.1298 \cdot 10^{-3}$	$-3.1035 \cdot 10^{-2}$	$-1.6287 \cdot 10^{-2}$
0.073752	0.063751	0.046243	0.073754	0.063773	0.046255	$-4.8732 \cdot 10^{-3}$	$-3.6619 \cdot 10^{-2}$	$-1.0104 \cdot 10^{-2}$
0.081336	0.071349	0.038739	0.081299	0.071332	0.038753	$-6.0656 \cdot 10^{-2}$	$-1.1496 \cdot 10^{-1}$	$-8.6065 \cdot 10^{-3}$
0.081259	0.071102	0.061205	0.081259	0.071294	0.061275	$-1.0671 \cdot 10^{-2}$	$-6.2310 \cdot 10^{-2}$	$-4.0899 \cdot 10^{-2}$
0.073750	0.063775	0.053756	0.073752	0.063789	0.053765	$-3.2353 \cdot 10^{-3}$	$-6.1994 \cdot 10^{-2}$	$-2.7847 \cdot 10^{-2}$
0.066252	0.056255	0.046253	0.066252	0.056259	0.046255	$-3.0596 \cdot 10^{-3}$	$-1.7102 \cdot 10^{-2}$	$-1.0054 \cdot 10^{-2}$
0.058749	0.048703	0.038729	0.058745	0.048732	0.038742	$8.2877 \cdot 10^{-3}$	$3.6864 \cdot 10^{-2}$	$2.0385 \cdot 10^{-2}$

Tabla VII.4.13: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{21} y k_{23} cuando hemos tomado como estimaciones iniciales los valores de ANN del modelo **I**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
0.058750	0.048750	0.061250	0.059328	0.0011356	0.026483	-0.98302	97.670	56.763
0.066250	0.056250	0.053750	0.064869	-0.0025079	0.020803	2.0839	104.46	61.296
0.073750	0.063750	0.046250	0.069575	-0.0073313	0.015470	5.6611	111.50	66.550
0.081250	0.071250	0.038750	0.073259	-0.0143078	0.010470	9.8357	120.08	72.980
0.081250	0.071250	0.061250	0.077989	-0.0032475	0.022999	4.0128	104.56	62.451
0.073750	0.063750	0.053750	0.070884	-0.0041495	0.019745	3.8854	106.51	63.266
0.066250	0.056250	0.046250	0.063785	-0.0054046	0.016441	3.7208	109.61	64.451
0.058750	0.048750	0.038750	0.056686	-0.0072306	0.013086	3.5127	114.83201	66.229

Tabla VII.4.14: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{23} y k_{32} del modelo **II**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
0.058750	0.048750	0.061250	0.059330	0.00066181	0.026478	-0.98797	98.642	56.771
0.066250	0.056250	0.053750	0.064882	-0.0016197	0.020772	2.0656	102.88	61.355
0.073750	0.063750	0.046250	0.069650	-0.0052538	0.015315	5.5591	108.24	66.886
0.081250	0.071250	0.038750	0.073451	-0.011363	0.010145	9.5989	115.95	73.820
0.081250	0.071250	0.061250	0.078026	-0.0021670	0.022931	3.9679	103.04	62.561
0.073750	0.063750	0.053750	0.070922	-0.0028036	0.019666	3.8343	104.40	63.413
0.066250	0.056250	0.046250	0.063829	-0.0037348	0.016335	3.6531	106.64	64.682
0.058750	0.048750	0.038750	0.056738	-0.0051152	0.012956	3.4241	110.49	66.566

Tabla VII.4.15: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{23} y k_{31} del modelo **III**.

2) En una segunda etapa se lleva a cabo el tratamiento de los datos de absorbancia/tiempo procedentes del mecanismo **II** considerando que los mecanismos **I**, **II** y **III** pudieran ser responsables del transcurso de la reacción. Se procede de igual manera que en el caso anterior y a la vista de los resultados (Tablas VII.4.16, VII.4.17 y VII.4.18) se selecciona el mecanismo que mejor representa a los datos de absorbancia, modelo **II**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
0.058750	0.048746	0.061248	0.058750	0.048748	0.061249	$-4.8000 \cdot 10^{-4}$	$4.8903 \cdot 10^{-3}$	$2.4457 \cdot 10^{-3}$
0.066250	0.056249	0.053750	0.066250	0.056249	0.053749	$-3.2151 \cdot 10^{-4}$	$1.6907 \cdot 10^{-3}$	$1.8977 \cdot 10^{-4}$
0.073749	0.063751	0.046251	0.073750	0.063749	0.046249	$-5.3288 \cdot 10^{-4}$	$2.0282 \cdot 10^{-3}$	$2.1557 \cdot 10^{-3}$
0.081250	0.071255	0.038749	0.081251	0.071251	0.038749	$-7.1262 \cdot 10^{-4}$	$-7.7474 \cdot 10^{-4}$	$3.1458 \cdot 10^{-3}$
0.081249	0.071253	0.061255	0.081249	0.071252	0.061254	$1.0646 \cdot 10^{-3}$	$-2.2540 \cdot 10^{-3}$	$-5.9102 \cdot 10^{-3}$
0.073750	0.063750	0.053749	0.073751	0.063745	0.053746	$-1.6393 \cdot 10^{-3}$	$7.3663 \cdot 10^{-3}$	$7.0847 \cdot 10^{-3}$
0.066249	0.056254	0.046245	0.066250	0.056251	0.046247	$-2.9736 \cdot 10^{-4}$	$-9.9378 \cdot 10^{-4}$	$7.3492 \cdot 10^{-3}$
0.058748	0.048727	0.038748	0.058749	0.048738	0.038749	$1.3396 \cdot 10^{-3}$	$2.4441 \cdot 10^{-2}$	$2.5884 \cdot 10^{-3}$

Tabla VII.4.16: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{23} y k_{32} cuando hemos tomado como estimaciones iniciales los valores de ANN del modelo **II**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
0.058750	0.048750	0.061250	-	-	-	-	-	-
0.066250	0.056250	0.053750	0.077173	-0.068946	-0.017217	-16.488	222.57	132.03
0.073750	0.063750	0.046250	0.087471	-0.079599	-0.019675	-18.605	224.86	142.54
0.081250	0.071250	0.038750	-	-	-	-	-	-
0.081250	0.071250	0.061250	0.097483	-0.072869	-0.013141	-19.979	202.27	121.45
0.073750	0.063750	0.053750	0.087330	-0.073128	-0.016224	-18.414	214.71	130.18
0.066250	0.056250	0.046250	0.077299	-0.075787	-0.020976	-16.679	234.73	145.35
0.058750	0.048750	0.038750	-	-	-	-	-	-

Tabla VII.4.17: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{21} y k_{23} del modelo **I**.

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
0.058750	0.048750	0.061250	0.061224	0.037857	0.021022	-4.2109	22.344	65.678
0.066250	0.056250	0.053750	0.068966	0.045285	0.018975	-4.1003	19.494	64.698
0.073750	0.063750	0.046250	0.076563	0.053156	0.016763	-3.8144	16.618	63.755
0.081250	0.071250	0.038750	0.084043	0.061369	0.014379	-3.4370	13.868	62.892
0.081250	0.071250	0.061250	0.085616	0.055466	0.020851	-5.3739	22.153	65.957
0.073750	0.063750	0.053750	0.077036	0.051311	0.018888	-4.4560	19.513	64.860
0.066250	0.056250	0.046250	0.068575	0.046943	0.016873	-3.5087	16.546	63.519
0.058750	0.048750	0.038750	0.060288	0.042165	0.014871	-2.6174	13.507	61.623

Tabla VII.4.18: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{23} y k_{31} del modelo III).

3) Por último se han generado datos cinéticos para el mecanismo III y después de haber realizado el mismo proceso que en los casos anteriores se concluye que el mecanismo que mejor representa los datos cinéticos es el III (Tablas VII.4.19, VII.4.20 y VII.4.21).

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{31}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{31})
0.058749	0.048749	0.061246	0.058748	0.048754	0.061248	$2.8800 \cdot 10^{-3}$	$-7.9097 \cdot 10^{-3}$	$3.7894 \cdot 10^{-3}$
0.066250	0.056249	0.053751	0.066247	0.056258	0.053750	$3.9109 \cdot 10^{-3}$	$-1.3851 \cdot 10^{-2}$	$-4.2791 \cdot 10^{-5}$
0.073750	0.063749	0.046249	0.073747	0.063758	0.046250	$4.3485 \cdot 10^{-3}$	$-1.3062 \cdot 10^{-2}$	$-8.4324 \cdot 10^{-5}$
0.081255	0.071247	0.038732	0.081242	0.071269	0.038750	$1.0055 \cdot 10^{-2}$	$-2.6187 \cdot 10^{-2}$	$-9.6516 \cdot 10^{-4}$
0.081252	0.071254	0.061252	0.081233	0.071297	0.061248	$2.1420 \cdot 10^{-2}$	$-6.5593 \cdot 10^{-2}$	$3.5233 \cdot 10^{-3}$
0.073750	0.063749	0.053750	0.073746	0.063759	0.053749	$4.9912 \cdot 10^{-3}$	$-1.5187 \cdot 10^{-2}$	$3.3860 \cdot 10^{-4}$
0.066250	0.056250	0.046249	0.066247	0.056257	0.046250	$4.4362 \cdot 10^{-3}$	$-1.2668 \cdot 10^{-2}$	$-1.3622 \cdot 10^{-4}$
0.058749	0.048749	0.038752	0.058749	0.048749	0.038751	$1.0757 \cdot 10^{-3}$	$5.1282 \cdot 10^{-4}$	$-3.7032 \cdot 10^{-3}$

Tabla VII.4.19: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{23} y k_{31} cuando hemos tomado como estimaciones iniciales los valores de ANN del modelo III).

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	k_{12}/min^{-1}	k_{21}/min^{-1}	k_{23}/min^{-1}	Dev % (k_{12})	Dev % (k_{21})	Dev % (k_{23})
0.058750	0.048750	0.061450	-	-	-	-	-	-
0.066250	0.056250	0.053750	-	-	-	-	-	-
0.073750	0.063750	0.046250	0.090043	-0.052468	-0.0088579	-22.092	182.30	119.15
0.081250	0.071250	0.038750	0.10026	-0.062587	-0.0098174	-23.400	187.84	125.34
0.081250	0.071250	0.061250	-	-	-	-	-	-
0.073750	0.063750	0.053750	0.90517	-0.047351	-0.0076605	-1127.4	174.28	114.25
0.066250	0.056250	0.046250	0.079439	-0.048575	-0.0095451	-19.908	186.36	120.64
0.058750	0.048750	0.038750	0.068685	-0.050689	-0.012203	-16.911	203.98	131.49

Tabla VII.4.20: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{21} y k_{23} del modelo I).

Valores - Iniciales			Valores - Finales			Desviación		
k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	k_{12}/min^{-1}	k_{23}/min^{-1}	k_{32}/min^{-1}	Dev % (k_{12})	Dev % (k_{23})	Dev % (k_{32})
0.058750	0.048750	0.061450	0.068553	0.031193	0.18401	-16.686	36.015	-200.43
0.066250	0.056250	0.053750	0.075719	0.044353	0.16349	-14.293	21.150	-204.18
0.073750	0.063750	0.046250	0.081543	0.060777	0.14852	-10.567	4.6639	-221.13
0.081250	0.071250	0.038750	0.086195	0.076649	0.12672	-6.0857	-7.5787	-227.02
0.081250	0.071250	0.061250	0.095526	0.053636	0.18593	-17.571	24.721	-203.57
0.073750	0.063750	0.053750	0.084089	0.053848	0.16955	-14.018	15.533	-215.44
0.066250	0.056250	0.046250	0.072983	0.053367	0.14975	-10.164	5.1252	-223.79
0.058750	0.048750	0.038750	0.061976	0.052649	0.12823	-5.4902	-7.9996	-230.92

Tabla VII.4.21: Resultados del proceso de *Predicción* obtenidos después de la aplicación AH para k_{12} , k_{23} y k_{32} del modelo **II**.

De todo lo expuesto en el apartado VII.4 (desglosado en los subapartados VII.4.1 y VII.4.2) se constata que el algoritmo híbrido AH (ANN-AGDC) posee la capacidad de discriminar ó distinguir entre varios modelos ó mecanismos de reacción distintos, aunque muy similares, que pudieran ser los responsables del transcurso de una reacción química. Además, AH es capaz de decidir cuál de ellos es más probable, pudiendo determinar al mismo tiempo, los valores de las constantes cinéticas individuales de cada etapa, lo cual supone una gran ventaja en el campo de la *Modelización* en Cinética Química.

*El éxito no nos lo brinda el reconocimiento ajeno.
Es el resultado de aquello que plantaste con amor.
Cuando llega el momento de cosechar, puedes decir:
“Lo he conseguido”
El manuscrito encontrado en Accra*

VIII. Conclusiones

Capítulo VIII.

Conclusiones

1. Del apartado VII.2:

- La aplicación del nuevo algoritmo híbrido AH (ANN-AGDC) a cinéticas no isotérmicas permite determinar Parámetros Termodinámicos de Activación (PTA), directamente, sin necesidad de determinar las constantes cinéticas en un paso previo.
- En el caso de experimentos no isotérmicos, solamente hace falta una única cinética replicada, para obtener un gran colectivo de valores experimentales, mientras que el colectivo de datos obtenido de las cinéticas isotérmicas es más reducido y el número de experiencias necesarias para conseguir dichos valores es más elevada. Por lo tanto, con la aplicación de las cinéticas no isotérmicas se consigue un mayor número de datos siendo suficiente la realización de menos experimentos en el laboratorio, lo cual conlleva un ahorro considerable de tiempo y reactivos.
- De todo esto podemos concluir que el tratamiento de datos cinéticos no isotérmicos presenta grandes ventajas metodológicas experimentales sobre los métodos clásicos isotérmicos.
- En las cinéticas clásicas una vez que se ha calculado el valor de la constante cinética, es necesario llevar a cabo la linealización de la ecuación de Arrhenius y/o Eyring para obtener los PTA. Esto implica además una transformación matemática de las ecuaciones y de los datos experimentales que produce una modificación, tanto de sus errores intrínsecos como de los errores y precisiones en los valores de los parámetros de activación obtenidos.
- Este problema no existe cuando se utiliza AH (ANN-AGDC). Si bien el tratamiento de datos es mucho más complejo, se puede aplicar perfectamente al ser un método riguroso y robusto, como nos demuestran los resultados que hemos obtenido.
- Tras analizar y comparar los resultados mostrados en las diversas figuras del Capítulo VII hemos observado que la representación gráfica de los valores obtenidos conjuntamente para ambos parámetros tras la aplicación de ANN proporciona una correlación aparentemente aceptable a primera vista. Sin embargo, cuando hemos representado cada parámetro de manera individual en la escala adecuada, se observa una gran dispersión de los puntos en la mayoría de los casos, concluyendo que la aplicación de ANN exclusivamente es inaceptable. Dicho motivo ha justificado la necesidad de la aplicación posterior del algoritmo de

optimización AGDC en un segundo paso para mejorar los resultados obtenidos con ANN y que ha dado lugar al nuevo algoritmo híbrido AH.

- La gran ventaja de ANN es que es un tratamiento que no necesita estimaciones iniciales. Por este motivo se aplica inicialmente ANN, de tal forma que los resultados obtenidos pueden ser utilizados como valores para las estimaciones iniciales en el segundo método (AGDC) que sí requiere el uso de valores previos para los parámetros a determinar.
- La aplicación de AGDC posee otra ventaja evidente que sirve para refinar los valores obtenidos previamente en ANN hasta conseguir los valores óptimos de los PTA.
- De todo lo anterior sacamos una importante conclusión según la cual la aplicación individual de los métodos ANN y AGDC proporciona resultados no aceptables o insuficientes. Sin embargo, la aplicación secuencial de ambos métodos (Algoritmo Híbrido, AH) aporta un conjunto de resultados óptimos.
- Observando las Figuras de los Residuales del Capítulo VII se comprueba fácilmente que los valores de los Residuales obtenidos con ANN son mucho más elevados que los obtenidos con el nuevo Algoritmo Híbrido (AH). Esto nos ratifica que la aplicación de los dos métodos secuencialmente ofrece resultados buenos y satisfactorios para la determinación de los valores optimizados de los Parámetros Termodinámicos de Activación.

A la vista de todos los resultados obtenidos en el Capítulo VII (Apartado VII.2), podemos concluir que el Algoritmo Híbrido AH que proponemos es un método computacional robusto para el tratamiento de datos cinéticos no isotérmicos y para la determinación de los valores de los PTA en los sistemas tratados. Estos buenos resultados nos permiten proponerlo como un método muy útil y eficaz para el tratamiento de datos cinéticos experimentales adquiridos en el laboratorio.

2. Del apartado VII.3:

- Los valores de las constantes cinéticas (k_{12} y k_{23}) del sistema $B_1 \xrightarrow{k_{12}} B_2$ que han sido *predecidos* mediante ANN a partir de los datos cinéticos obtenidos a las 3 primeras λ (350, 375 y 400 nm), tienen valores similares. Sin embargo, los valores *predichos* a partir de datos cinéticos obtenidos a 450 nm, son discordantes con los anteriores, con valores ciertamente diferentes de ambas constantes cinéticas. En consecuencia, estos valores obtenidos a partir de datos cinéticos medidos a 450 nm no deben intervenir en el cálculo del valor promedio de ambas constantes cinéticas ($\langle k_{12} \rangle$ y

$\langle k_{23} \rangle$) que hemos de asignar como valores reales y únicos a la reacción química objeto de estudio cinético.

- Por tanto las 3 primeras λ , (350, 375 y 400 nm) son válidas para realizar el estudio cinético de la reacción mientras que $\lambda=450$ nm, no es una longitud de onda adecuada para el tratamiento cinético de la reacción química en estudio.
 - Los valores medios de ambas constantes cinéticas (k_{12} y k_{23}), calculados únicamente con los datos cinéticos medidos a las 3 primeras λ (descartando $\lambda=450$ nm), admitiendo los errores y precisiones descritos en la bibliografía, son: $\langle k_{12} \rangle = 0.338_1 \text{ min}^{-1}$ y $\langle k_{23} \rangle = 0.019_5 \text{ min}^{-1}$. Estos valores se los asignamos a la cinética de la reacción en estudio como valores únicos determinados mediante tratamiento computacional ANN. Esto es lógico, pues a 450 nm las especies B_1 y B_3 no absorben luz, es decir, poseen valores nulos de los coeficientes de absorción molar y al mismo tiempo, el valor del correspondiente a B_2 , es pequeño. Dicho de otra forma, a 450 nm solo hay aportación al valor de la absorbancia total de la mezcla por parte de la especie intermedia B_2 , lo cual representa una información cinética insuficiente para la correcta obtención de ambas constantes cinéticas. Esto puede corroborarse observando la Figura VII.3.5, en que el perfil de esta cinética tiene un intervalo de variación de absorbancia muy pequeño (desde 0 al tiempo inicial a 0.1 en el máximo) que la inhabilita para un tratamiento cinético riguroso.
 - Estas conclusiones coinciden, con las existentes en bibliografía (ver Tabla VII.3.1, 2ª y 3ª columnas), aportadas por varios grupos de investigadores que han estudiado de forma independiente y en distintas épocas la cinética de esta reacción química, si bien mediante la aplicación de métodos cinéticos clásicos.
3. Del apartado VII.4:
- Con la técnica de ANN bajo la modalidad *Soft modelling*, hemos podido comprobar la valiosa posibilidad de establecer el modelo más probable al que los datos cinéticos experimentales pertenecen, ya que la metodología cinética tiene la capacidad de discriminación entre los distintos mecanismos o modelos que se consideran aplicables a una reacción química.
 - El algoritmo AH (ANN-AGDC) discrimina entre 3 modelos distintos en el que intervienen el mismo número de especies, de reacciones entre ellas y de constantes de velocidad. Hay que resaltar que la aplicación exclusiva de ANN consigue realizar con éxito esta misma discriminación.
 - Los modelos cinéticos son diferentes en cuanto a su arquitectura (lineal o cíclica) o bien en cuanto a la distribución de las reacciones individuales o constantes

cinéticas, (3 especies y 3 constantes cinéticas) pero en todos ellos hemos podido llevar a cabo el tratamiento con el mismo ED obteniendo excelentes resultados.

- Por tanto podemos concluir, que tanto el método ANN como el nuevo algoritmo híbrido AH son capaces de distinguir o discriminar entre modelos diferentes en el proceso de *Predicción* conduciendo a determinación de valores correctos de constantes de velocidad individuales de cada etapa, únicamente cuando se realiza el tratamiento de los datos cinéticos pertenecientes al modelo que sirvió para llevar a cabo su *Entrenamiento*. En caso contrario, cuando no pertenece a dicho modelo, el proceso de *Predicción* fracasa, obteniendo valores absurdos e inaceptables. Este hecho permite decidir cuál de todos los modelos puede ser el responsable del transcurso de la reacción química en estudio, lo cual supone una gran ventaja en el campo de la Modelización en Cinética Química.

*Quien presta atención a su día,
Descubre el instante mágico.
A orillas del río piedra me senté y lloré*

IX. Referencias Bibliográficas

Capítulo IX.

Referencias Bibliográficas

- (1) Yao, R.; Yang, B.; Cheng, G.; Tao, X.; Meng, F. “Kinetics research for the synthesis of branch ether using genetic-simulated annealing algorithm with multi-pattern evolution” *Chemical Engineering Journal* **2003**, *94*,113-119.
- (2) Maeder, M.; Neuhold, Y. M.; Puxty, G. “Application of a genetic algorithm: Near optimal estimation of the rate and equilibrium constants of complex reaction mechanisms” *Chemom. Intell. Lab. Syst.* **2004**, *70*, 193-203.
- (3) Saha, B.; Karthik Reddy, P.; Ghoshal, A.K. “Hybrid genetic algorithm to find the best model and the globally optimized overall kinetics parameters for thermal decomposition of plastics” *Chemical Engineering Journal* **2008**,*138*, 20-29.
- (4) Dai, K.; Wang, N. “A Hybrid DNA based genetic algorithm for parameter estimation of dynamic systems” *Chemical Engineering Research and Design* **2012**, *90*, 2235-2246.
- (5) Elhosseini, M.A.; Haikal, A.Y. “Modified hybrid algorithm for process optimization” *Applied Mathematics and Computation* **2014**, *232*, 1209-1224.
- (6) Niazi, A.; Leardi, R. “Genetic Algorithms in Chemometrics” *J. Chemom.* **2012**, *26* 345-351.
- (7) Hervás, C.; Algar, J. A.; Silva, M. “Correction of temperature variations in kinetic based. Determinations by use of pruning computational neural networks in conjunction with genetic algorithms” *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 724–731.
- (8) Papes Filho, A.C.; Maciel Filho, R. “Hybrid training approach for artificial neural networks using genetic algorithms for rate of reaction estimation: Application to industrial methanol oxidation to formaldehyde on silver catalyst” *Chemical Engineering Journal* **2010**,*157*, 501-508.
- (9) Huang, M.; Ma, Y.; Wan, J.; Zhang, H.; Wang, Y.; Chen, Y.; Yoo, C.; Guo, W. “A hybrid genetic-Neural algorithm for modeling the biodegradation process of BnBP in AAO system” *Bioresource Technology* **2011**,*102*, 8907-8913.
- (10) Lisa, G.; Apreutesei Wilson, D.; Curteanu, S.; Lisa, C.; Piuleac, C.P.; Bulacovschi, V. “Ferrocene derivatives thermostability prediction using neural networks and genetic algorithms” *Thermochica Acta* **2011**, *521*, 26-36.

-
- (11) Schubert, M.; Muffler, A.; Mourad, S. "The use of a radial basis neural network and genetic algorithm for improving the efficiency of laccase-mediated dye decolourization" *Journal of Biotechnology* **2012**, *161*, 429-436.
- (12) Karimi, H.; Ghaemi, M. "Application of artificial neural network and genetic algorithm to modeling and optimization of removal of methylene blue using activated carbon" *Journal of Industrial and Engineering Chemistry* **2014**, *20*, 2471-2476.
- (13) Assefi, P.; Ghaedi, M.; Ansari, A.; Habibi, M.H.; Momeni, M.S. "Artificial neural network optimization for removal of hazardous dye Eosin Y from aqueous solution using Co_2O_3 -NP-AC: Isotherm and kinetics study" *Journal of Industrial and Engineering Chemistry* **2014**, *20*, 2905-2913.
- (14) Ghaedi, M.; Ansari, A.; Bahari, F.; Ghaedi, A.M.; Vafaei, A. "A Hybrid artificial neural network and particle swarm optimization for prediction of removal of hazardous dye brilliant green from aqueous solution using zinc sulfide nanoparticle loaded on activated carbon" *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* **2015**, *137*, 1004-1015.
- (15) Maghsoudi, M.; Ghaedi, M.; Zinali, A.; Ghaedi, A.M.; Habibi, M.H. "Artificial neural network (ANN) method for modeling of sunset yellow dye adsorption using zinc oxide nanorods loaded on activated carbon: Kinetic and isotherm study" *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy* **2015**, *134*, 1-9.
- (16) Freeman A.; y D.M. Skapura D.M. "*Neural Networks, Algorithms, Applications and Programming Techniques*", Adinson-Wesley, Publishing Company, Inc. Reading, Massachusetts (1991).
- (17) Zupan, J.; Gasteiger, J. "*Neural Networks for Chemists*", VCH, Weinheim, Germany (1993).
- (18) Vandeginste, G.M.; Massart, D.L.; Buydens, L.M.C.; Jong, S.; Lewi, P.J.; Smeyeres-Verbeke, J. "*Handbook of Chemometrics and Qualimetrics*", Ed. ELSEVIER. Amsterdam, (1998).
- (19) Ramis Ramos, G.; García, M.C. "*Quimiometría*", Ed. Síntesis, Madrid (2001)
- (20) Ni, Y.; Liu, C.; Kokot, S. "Simultaneous kinetic-spectrophotometric determination of acetaminophen and phenobarbital by artificial neural networks and partial least squares" *Analytica Chimica Acta* **2000**, *419*, 185-196.
- (21) Ni, Y.; Wang, Y.; Kokot, S. "Multicomponent Kinetic Spectrophotometric Determination of Pefloxacin and Norfloxacin in Pharmaceutical Preparations and

- Human Plasma Samples with the Aid of Chemometrics,” *Spectrochica Acta* **2008**, *70*, 1049-1059.
- (22) Goicoechea, H.C.; Collado, M.S.; Satuf, M.L.; Olivieri, A.C. “Complementary use of partial least-squares and artificial neural networks for the non-linear spectrophotometric analysis of pharmaceutical samples” *Anal. Bioanal. Chem.* **2002**, *374*, 460-465.
- (23) Havel, J. ; Lubal, P. ; Farková, M. « Evaluation of chemical equilibria with the use of artificial neural” *Polyhedron* **2002**, *21*,1375-1384.
- (24) Boozarjomehry, R. B.; Abdolahi, F.; Moosavian, M. A. “Characterization of basic properties for pure substances and petroleum fractions by neural networks” *Fluid Phase Equilibria* **2005**, *231*,188-196.
- (25) Bezerra, E.M. ; Bento, M.S. ; Rocco, J.A.F.F. ; Iha, K. ; Lourenço, V.L. ; Pardini, L.C. “Artificial neural network (ANN) prediction of kinetic parameters of (CRFC) composites” *Computacional Materials Science* **2008**, *44*, 656-663.
- (26) Dutot, A.L.; Rude, J. ; Aumont, B. “Neural network method to estimate the aqueous rate constants for the OH reactions with organic compounds” *Atmospheric Environment* **2003**, *37*, 269–276.
- (27) Urata, S.; Takada, A.; Uchimar, T.; Chandra, A.K. “Rate constants estimation for the reaction of hydrofluorocarbons and hydrofluoroethers with OH radicals” *Chemical Physics Letters* **2003**, *368*, 215-223
- (28) Ventura, S.; Silva, M.; Pérez-Bendito, D.; Hervás, C. “Computational Neural Networks in Conjunction with Principal Component Analysis for Resolving Highly Nonlinear Kinetics” *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 287-291.
- (29) Ventura, S.; Silva, M.; Pérez-Bendito, D. Hervás, C. “Estimation of Parameters of Kinetic Comprtmental Models by Use os Computational Neural Networks” *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 517-521
- (30) Hervás, C.; Ventura, S.; Silva, M.; Pérez-Bendito, D. “Computational Neural Networks for Resolving Nonlinear Multicomponent Systems Based on Chemiluminescence Methods” *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1119-1124.
- (31) Blanco, M.; Coello, J.; Iturriaga, H.; Maspoeh, S.; Porcel, M. “Use of circular dichroism and artificial neural networks for the kinetic-spectrofotometric resolution of enantiomers” *Analytica Chimica Acta* **2001**, *431*, 115-123.
- (32) López-Cueto, G.; Ostra, M.; Ubide, C. “Linear and non-linear multivariate calibration methods for multicomponent kinetic determinations in cases of severe non-linearity” *Analytica Chimica Acta* **2000**, *405*, 285-295.

- (33) Ni, Y.; Huang, C.; Kokot, S. "A kinetic-spectrophotometric method for the determination of ternary mixtures of reducing sugars with the aid artificial neural networks and multivariate calibration" *Analytica Chimica Acta* **2003**, *480*, 53-65.
- (34) Ni, Y.; Huang, C.; Kokot, S. "Application of multivariate calibration and artificial neural networks to simultaneous kinetic-spectrophotometric determination of carbamate pesticides" *Chemom. Intell. Lab. Syst.* **2004**, *71*, 177-193.
- (35) Ni, Y.; Huang, C.; Kokot, S. "Simultaneous determination of iron and aluminium by differential kinetic spectrophotometric method and chemometrics" *Analytica Chimica Acta* **2007**, *599*, 209-218.
- (36) Ni, Y.; Wang, Y. "Application of chemometric methods to the simultaneous kinetic spectrophotometric determination of iodate and periodate based on consecutive reactions," *Microchemical Journal* **2007**, *86*, 216-226.
- (37) Ni, Y.; Wang, Y.; Kokot, S. "Simultaneous kinetic-spectrophotometric determination of maltol and ethyl maltol in food samples by using chemometrics" *Food Chemistry* **2008**, *109*, 431-438.
- (38) Ni, Y.; Wang, Y.; Kokot, S. "Simultaneous kinetic spectrophotometric analysis of five synthetic food colorants with aid of chemometrics" *Talanta* **2009**, *78*, 432-441.
- (39) Safavi, A.; Absalan, G.; Maesum, S. "Simultaneous determination of V(IV) and Fe(II) using "neural networks" through a single catalytic kinetic run" *Analytica Chimica Acta* **2001**, *432*, 229-233.
- (40) Safavi, A.; Abdollahi, H.; Hormozi Nezhad, M.R. "Artificial neural networks for simultaneous spectrophotometric differential kinetic determination of Co(II) and V(IV)" *Talanta* **2003**, *59*, 515-523.
- (41) Chamsaz, M.; Safavi, A.; Fadaee, J. "Simultaneous kinetic-spectrophotometric determination of carbidopa, levadopa and methyl dopa in the presence of citrate with the aid of multivariate calibration and artificial neural networks" *Analytica Chimica Acta* **2007**, *603*, 140-146.
- (42) Ren, S.; Gao, L. "Wavelet packet transform and artificial neural networks to simultaneous kinetic multicomponent determination" *Anal. Bioanal. Chem.* **2004**, *378*, 1392-1398.
- (43) Magni, D.M.; Olivieri, A.C.; Bonivardi, A.L. "Artificial neural networks study of the catalytic reduction of resazurin : stopped-flow injection kinetic-spectrophotometric determination of Cu(II) and Ni(II) " *Analytica Chimica Acta* **2005**, *528*, 275-284.
- (44) Chen, X.; Xu, H.; Dong, L.; Liu, H.; Zeng, Y.; Hu, Z. "Application of artificial neural networks in multivariable optimization of an on-line microwe FIA system for catalytic kinetic determination of iridium(III)" *Anal. Bioanal. Chem.* **2002**, *373*, 883-888.

-
- (45) Molga, E.J.; Wozik, B.A.A.; Westerterp, K.R. "Neural networks for modelling of chemical reaction systems with kinetics: oxidation of 2-octanol with nitric acid" *Chemical Engineering and Processing* **2000**, *39*, 323-334.
- (46) Serra, J. M.; Corma, A.; Argente, E.; Valero, S.; Botti, V. "Neural networks for modeling of kinetic reaction data applicable to catalyst scale up and process control and optimization in the frame combinatorial catalysis" *Applied Catalysis A: General* **2003**, *254*, 133-145.
- (47) Rönholm, M.; Klingstedt, F.; Eränen, K.; Lindfors, L.E. "Artificial neural networks modeling applied to NO_x reduction with octane in excess oxygen over Ag/Al₂O₃" *React. Kinet. Catal. Lett.* **2003**, *78*, 331-340.
- (48) Ardakani, S. J.; Gobal, F. "On the rate oxidation of CO on La₂O₃ doped NiO/Al₂O₃ catalysts: An Artificial Neural Networks approach" *React. Kinet. Catal. Lett.* **2005**, *85*, 347-353.
- (49) Valeh-e-Sheyda, P.; Yaripour, F.; Moradi, G.; Saber, M. "Application of Artificial Neural Networks Estimation of Reaction Rate in Methanol Dehydration" *Ind. Eng. Chem. Res.* **2010**, *49*, 4620-4626.
- (50) Szaleniec, M.; Witko, M.; Tadeusiewicz, R.; Goclon, J. "Application of artificial neural networks and DFT-based parameters for prediction of reaction kinetic of ethylbenzene dehydrogenase" *J. Comput. Aided Mol. Des.* **2006**, *20*, 145-157.
- (51) Ni, Y.; Cao, D.; Kokot, S. "Simultaneous enzymatic kinetic determination of pesticides, carbaryl and phoxim, with the aid of chemometrics" *Analytica Chimica Acta* **2007**, *588*, 131-139.
- (52) Bas, D.; Boyaci, I. H. "Modeling and optimization I: Usability of response surface methodology" *Journal of Food Engineering* **2007**, *78*, 846-854.
- (53) Bas, D.; Dudak, F. C.; Boyaci, I. H. "Modeling and optimization III: Reaction rate estimation using artificial neural network (ANN) without a kinetic model" *Journal of Food Engineering* **2007**, *79*, 622-628.
- (54) Bas, D.; Dudak, F. C.; Boyaci, I. H. "Modeling and optimization IV: Investigation of reaction kinetics and kinetic constants using a program in which artificial neural network (ANN) was integrated" *Journal of Food Engineering* **2007**, *79*, 1152-1158.
- (55) Silva, J. A.; Holanda C. Neto, E.; Adriano, W.S.; Ferreira, A.L.O.; Gonçalves, L. R. B. "Use of neural network in the mathematical modeling of the enzymic synthesis of amoxicillin catalyzed by penicillin G acylase immobilized in chitosan" *World J Microbiol Biotechnol.* **2008**, *24*, 1761-1767.
- (56) Kovács, B.; Tóth, J. *World Academy of Science*, "Estimating Reaction Rate Constants with Neural Networks" *Engineering and Technology* **2007**, *26*, 13-17.

-
- (57) Halberstam, N.M.; Baskin, I. I.; Palyulina, V. A.; Zefirov, N. S. "Quantitative structure-conditions-property relationship studies. Neural network (modeling of the acid hydrolysis of esters" *Mendeleev Commun.* **2002**, *12*(5), 185–186.
- (58) Lemes, N. H. T.; Borges, E.; Braga, J. P. "Rate constants and absorption coefficients from experimental data: A inversion procedure based on recursive neural networks" *Chemom. Intell. Lab. Syst.* **2009**, *96*, 84-87.
- (59) Curteanu, S.; Cartwright, H. "Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks" *Journal Chemometrics* **2011**, *25*, 527-549.
- (60) Amato, F.; González-Hernández, J.L.; Havel, J. "Artificial neural networks combined with experimental design: A "soft" approach for chemical kinetics" *Talanta* **2012**, *93*, 72-78.
- (61) Trajan Neural Network Simulator, Release 3.0.D, Trajan Software Ltd., UK (1996-1998).
- (62) STATISTICA V.6, StatSoft Inc. USA.
- (63) González-Hernández, J.L.; Canedo, M. M.; Encinar, S. "Combining artificial neural networks and experimental design to prediction of kinetic rate constants" *J. Math. Chem.* **2013**, *51*, 1634-1653.
- (64) MATLAB & SIMULINK, © COPYRIGHT by The MathWorks, Inc. (1984–2014).
- (65) González-Hernández, J. L.; Canedo, M. M.; Domínguez-Gil, A.; Lanao, J. M. "Evaluation of Thermodynamic Ionization Constants by Computation of Spectrophotometric and Potentiometric Experimental Data Obtained Simultaneously from a Flow System" *J. Pharm. Sci.* **1992**, *81*, 592-596.
- (66) Luján, A.; González, J. L.; Canedo, M. M.; Grande, C. "Determination of Optimized Individual Kinetic Constants of the Aminolysis Reaction in Basic Medium of β -Lactam Antibiotics by numerical Computational Method" *J. Pharm. Sci.* **1993**, *82*, 1167-1171.
- (67) González-Hernández, J. L.; Canedo, M. M.; Grande, C. "Optimization of kinetic parameters. Multipurpose KINAGDC(MW) non-linear regression program" *Chemom. Intell. Lab. Syst.* **1997**, *39*, 77-84.
- (68) Canedo, M.M.; González-Hernández, J.L. "A new computational application of the AGDC algorithm for kinetic resolution of multicomponent mixtures (static and dynamic)" *Chemom. Intell. Lab. Syst.* **2003**, *66*, 63-78.
- (69) Canedo, M.M.; González-Hernández, J.L. "ANALKIN (AGDC): a multipurpose computational program for the kinetic resolution of multicomponent mixtures (static and dynamic)" *Chemom. Intell. Lab. Syst.* **2003**, *66*, 93-97.

-
- (70) González-Hernández, J. L.; Canedo, M. M.; Grande, C. "Classic and Multivariate Modeling Treatment of the Kinetics and Mechanism of Isomerization of 5-Cholesten-3-one Catalyzed by Sodium Ethoxide" *Int. J. Chem. Kinet.* **2006**, *38*, 38-47.
- (71) González-Hernández, J. L.; Canedo, M. M.; Grande, C. "Computational methods for the treatment of kinetics and mechanisms of the oxidation of 4-Cholesten-3-one by molecular oxygen by Sodium Ethoxide" *Int. J. Chem. Kin.* **2007**, *17(4)*, 289-302.
- (72) Canedo M. M.; González-Hernández J. L. "KINMODEL(AGDC): a multipurpose computational method for kinetic treatment" *J. Math. Chem.* **2011**, *49*, 163-184.
- (73) Canedo M. M.; González-Hernández J. L.; Encinar S. "Application de computational method KINMODEL(AGDC) to the simultaneous determination of kinetic and analytical parameters" *App. Math. and Comp.* **2013**, *219*, 7089-7101.
- (74) González-Hernández J. L.; Canedo M. M.; Grande, C. "Combining Different Mathematical Optimization Methods: A New "Hard-Modelling" Approach for Chemical Kinetic" *MATCH Commun. Math. Comput. Chem.* **2013**, *70*, 951-970.
- (75) Brown, E.; Phillpotts, C.A.R. "Non-Isothermal Kinetics" *Journal of Chemical Education* **1978**, *55 (9)*, 556-560.
- (76) González, J.L.; Salvador, F. "Kinetics of reactions in solution: method for the treatment of data from non-isothermal chemical kinetic experiments" *React. Kinet. Lett.* **1982**, *21(1-2)*, 167-171.
- (77) Salvador, F.; González, J.L. "Comparative study of non-isothermal methods in linear and non-linear temperature variation" *React. Kinet. Lett.* **1984**, *25(1-2)*, 125-130.
- (78) Salvador, F.; González, J.L.; Tel, L.M. "Non-Isothermic Chemical Kinetics in the Undergraduate Laboratory" *Journal of Chemical Education* **1984**, *61*, 921-922.
- (79) Alibrandi, G. "Non-isothermal spectrophotometric kinetics applied to inorganic reactions" *Inorganica Chimica Acta* **1994**, *221*, 31-34.
- (80) Maeder, M.; Molloy, K.J.; Schumacher, M.M. "Analysis of non-isothermal kinetic measurements" *Analytica Chimica Acta* **1997**, *337*, 73-81.
- (81) Hodgson, S.C.; Ngeh, L.N.; Orbell, J.D.; Bigger, S.W. "A student experiment in non-isothermal chemical kinetics" *Journal of Chemical Education* **1998**, *75(9)*, 1150-1153.
- (82) Wojciechowski, K.T.; Malecki, A. "SYMTERM - program for modelling chemical processes in non-isothermal conditions" *Computers and Chemistry* **2000**, *24*, 527-532.
- (83) Furusjö, E.; Svensson, O.; Danielsson, L.G. "Estimation of kinetic parameters from non-isothermal batch experiments monitored by in situ vibrational spectroscopy" *Chemometrics and Intelligent Laboratory Systems* **2003**, *66*, 1-14.
- (84) Maeder, M.; Molloy, K. J.; Schumacher, M. M. "Analysis of non-isothermal kinetic measurements" *Analytica Chimica Acta* **1997**, *337*, 73-81.

-
- (85) McCann, N.; Maeder, M. "Tutorial: The modelling of chemical processes" *Analytica Chimica Acta* **2009**, *647*, 31-39.
- (86) Rodríguez-Aragón, L.J.; López-Fidalgo, J. "Optimal designs for the Arrhenius equation" *Chemometrics and Intelligent Laboratory Systems* **2005**, *77*, 131-138.
- (87) Hong, J.; Guo, G.; Zhang, K. "Kinetics and mechanism of non-isothermal dehydration of nickel acetate tetrahydrate in air" *J. Anal. Pyrolysis* **2006**, *77*, 111-115.
- (88) Ribeiro, C.A.; de Souza, W.R.; Spirandeli Crespi, M.; Gomes Neto, J.A.; Fertoni, F.L. "Non-isothermal kinetic of oxidation of tungsten carbide" *Journal of Thermal Analysis and Calorimetry* **2007**, *90*(3), 801-805.
- (89) Vyazovkin, S.; Burnham, A.K.; Criado, J.M.; Pérez-Maqueda, L.A.; Popescu, C.; Sbirrazzuoli, N. "ICTAC Kinetics Committee recommendations for performing kinetic computations on thermal analysis data" *Thermochimica Acta* **2011**, *520*, 1-19.
- (90) Sesták, J. "The quandary aspects of non-isothermal kinetics beyond the ICTAC kinetic committee recommendations" *Thermochimica Acta* **2015**, *611*, 26-35.
- (91) Encinar, S.; González-Hernández, J. L.; Canedo, M. M. "A New Computational treatment in Non-isothermal Chemical Kinetics by Application of a Robust Hybrid Algorithm" *MATCH Commun. Math. Comput. Chem.* **2014**, *72*, 427-450
- (92) Encinar, S.; González-Hernández, J. L.; Canedo, M. M.; Juanes, D. "A robust hybrid algorithm (Neural Networks AGDC) applied to non-isothermal kinetics of consecutive chemical reactions" *J. Math. Chem.* **2015**, *53*, 1080-1104.
- (93) Tarek A.; El-Mihoub.; Hopgood A.; Nolle L.; Battersby A. "Hybrid Genetic Algorithms: A Review", *Engineering Letters* **2006**, *13*(2), 124-137.
- (94) Hilera J.R.; y Martínez V.J. "Redes Neuronales Artificiales", Alfaomega, Madrid, España (2000).
- (95) Kung S.Y. "Digital Neural Networks", PTR Prentice Hall, Inc. (1993).
- (96) Jackson P.C. "Introduction to Artificial Intelligence", Courier Dover Publications, Inc. New York (1985).
- (97) Kohonen T. "Self Organization and Associative Memory", Springer-Verlag, Berlín (1984).
- (98) Kohonen T. "An Introduction to Neural Computing", Ed. Neural Networks (1988).
- (99) Hetch-Wielsen R. "Theory of the Propagation Neural Network", Proc. Int. Joint Conf. (1989).
- (100) Corbacho J.M.; Díaz F.; Borrajo L.; Fernández F. "Redes Neuronales artificiales: Un enfoque práctico", Servicio de Publicacións da Universidade de Vigo (2000).
- (101) Mira J.; Delgado A.E.; Boticario J.G.; Díez F.J. "Aspectos Básicos de la Inteligencia Artificial", Sanz y Torres (2003).

-
- (102) Flórez López R.; Fernández Fernández J.M. “Las Redes Neuronales Artificiales: Fundamentos teóricos y aplicaciones prácticas”, Gesbiblo (2009).
- (103) Kosko B. “Bi-directional associative memories” *IEEE Transactions on Systems, Man and Cybernetics* **1987**, *18(1)*, 49-60.
- (104) Hebb D.O. “Organization of behaviour”, Science Editions, New York (1949).
- (105) Rosenblatt F. “Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms”, Spartan Books, Washington DC (1961).
- (106) McCulloch W.S.; W. Pitts. “A logical calculus of the ideas immanent in nervous activity”, *Bulleting of Mathematical Biophysics* **1943**, *5*, 115-133.
- (107) Widrow B.; Hoff M.E., “Adaptive switching circuits”, IRE WESCON Convention Record. New York **1960**, 96-104.
- (108) Widrow B.; Sterns S.D. “Adaptive Signal Processing”, Prentice-Hall, New York (1985).
- (109) Rumelhart D.E.; Hinton G.E.; Williams R.J. "Learning internal representations by error propagation", Eds., *Parallel Data Processing*, MA: The M.I.T. Cambridge **1986**, *1*, 318-362.
- (110) Rumelhart D.E.; Hinton G.E.; Williams R.J. "Learning representations by back – propagating errors”, *Nature*, **1986**, *323*, 533-536.
- (111) Hopfield J. "Neural Networks and physical systems with emergent collective computational abilities", *Proceedings of the national academy of Sciences* **1982**, *79*, 2554-2558.
- (112) Kosko B. “Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence”, Prentice-Hall, Englewood Cliffs, NJ (1992).
- (113) Grossberg S.; Levine D. “Adaptive pattern classification and universal recoding: Part I: Parallel development and coding of neural feature detectors”, *Biological Cybernetics*, **1976**, *23*, 121-134.
- (114) Grossberg S. “Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition and Motor Control”, Kluwer Academic, Norwell, Mass (1982).
- (115) Moreno M.N.; González J.L.; del Arco M.A.; Casado J. “Determination of macroscopic thermodynamic ionization constants under conditions of variable ionic strength by an optimization algorithm”, *Comput. Chem.*, **1990**, *14(2)*, 165-168.
- (116) Sánchez M.J. Memoria de Licenciatura, Departamento de Química Física, Universidad de Salamanca, **1995**.
- (117) Álvarez P. Memoria de Licenciatura, Departamento de Química Física, Universidad de Salamanca, **1995**.
- (118) Canedo M.M.; González J.L.; Álvarez P. CHEMOMETRICS IV. Celebrado en Brno (República Checa), Septiembre, **1996**.

-
- (119) Canedo M.M.; González J.L.; Álvarez P.; Grande C. "MICROTER-MW. A general program for the determination of the thermodynamics Microconstants of dissociation", *Int. J. Chem.* **1995**, 6(4), 115-122.
- (120) Canedo M.M.; Álvarez P.; González J.L. Programa MICROTER-MW Manual e instrucciones, Departamento de Química Física, Universidad de Salamanca, **1995**.
- (121) Canedo M.M.; Tesis Doctoral, Departamento de Química Física, Universidad de Salamanca, **1991**.
- (122) Canedo M.M.; González J.L.; Moreno M.N., "KINAGDC fitting to kinetic data by the AGDC algorithm", *Quantum Chemistry Program Exchange*, **1994**, 14(2),21.
- (123) Cuesta F. Memoria de Licenciatura, Departamento de Química Física, Universidad de Salamanca, **1991**.
- (124) Grande C. Memoria de Licenciatura, Departamento de Química Física, Universidad de Salamanca, **1993**.
- (125) Grande C.; Canedo M.M.; González J.L. "Kinetic Treatment of basic aminolysis of beta-lactam antibiotics by a univariatic optimization computational Method", *Int. J. Chem.* **1994**, 5(2), 77-85.
- (126) Grande C.; González J.L. "New Computational treatment of kinetic on the basic aminolysis of beta-lactam antibiotics by a joint uni-and multivariate optimization methods", *Int. J. Chem.*, **1994**, 5(3), 107-116.
- (127) Grande C.; González J.L. FRIS'94, Septiembre, celebrado en Burgos (España), **1994**.
- (128) Grande C.; Canedo M.M.; González J.L. "KINAGDC-MW: a multipurpose program for the treatment of kinetic and equilibrium data", *Comput. Chem.* **1996**, 20(2), 167-173.
- (129) Grande C. Tesis Doctoral, Departamento de Química Física, Universidad de Salamanca, **1998**.
- (130) Canedo M.M.; González J.L.; Casado M.R. "CONAGDC-MW. A computational program for the determination of concentrations in multicomponents mixtures, without previous separation or calibration", *Int. J. Chem.* **1999**, 9(1), 21-30.
- (131) Canedo M.M.; González J.L.; Casado M.R. "Simultaneous computational determination of kinetic constants and concentrations in multicomponents mixtures, without previous using the AGDC-MW algorithm", *Int. J. Chem.* **1999**, 9(2), 85-99.
- (132) Abdy, P. R.; Dempster, M. A. H. "Introduction to optimization Methods", Chapmanand Hall, Cambridge, (1974).
- (133) Wolfe, M. A. "Numerical Methods for unconstrained Optimization", Van Nostrand, 230, (1978).
- (134) Gill, P.; Murray, W.; Wright, M. H. "Practical Optimization", Academic Press, Inc., (1981).

-
- (135) Norris, A. C. "Computational Chemistry, An introduction to Numerical Methods", J. Wiley and Sons, Chichester, (1981).
- (136) Canedo M.M.; González J.L.; Cuesta F.; Havel J. "Computation in kinetics V. A new KINDC computational method. Comparative study with other computer programs for kinetic treatment", *React. Kinet. Catal. Lett.* **1992**, *48(1)*, 77-83.
- (137) Harwell Subroutine Library, "A Catalogue of Subroutines Computer Science and Systems Division". AERE Harwell, Oxfordshire. March 1984.
- (138) Gerald C.; Wheatley P.O. "Applied Numerical Analysis", Addison Wesley, pub. Co. (1984).
- (139) Mason C. "Métodos Matriciales", Ed. Anaya Multimedia, S.A., (1986).
- (140) Demidovich B.P.; Maron L.A. "Cálculo Numérico fundamental", Paraninfo (1977).
- (141) Puxty G.; Neuhold Y.M.; Jecklin; Ehly M.; Gemperline P.; Nordon A.; Littlejohn D.; Basford K.; De Cecco M; Hungerbühler K. 'Multivariate kinetic hard-modelling of spectroscopic data: A comparison of the esterification of butanol by acetic anhydride on different scales and with different instruments', *Chem. Eng. Sci.* **2008**, *63*, 4800-4809.
- (142) Kirsten J.M.; Maeder M.; Schumacher M. "Hard modelling of spectroscopic measurements. Applications in non-ideal industrial reaction systems" *Chemom. Intell. Lab. Syst.*, **1999**, *46*, 221-230.
- (143) Bijlsma S.; Smilde A.K. "Application of curve resolution based methods to kinetic data" *Anal. Chim. Acta.* **1999**, *396*, 231-240.
- (144) De Juan A.; Casassas E.; Tauler R. "Encyclopedia of Analytical Chemistry: Instrumentation and Applications. 'Soft-modelling of analytical data'", Wiley, New York (2000).
- (145) De Juan A.; Maeder M.; Martínez M.; Tauler R. "Combining hard- and soft-modelling to solve kinetic problems" *Chemom. Intell. Lab. Syst.* **2000**, *54*, 123-141.
- (146) Laidler K.J. "Symbolism and terminology in chemical kinetics", *Pure Appl. Chem.* **1981**, *53*, 753-771.
- (147) Laidler K.J. "A glossary of terms used in chemical kinetics, including reaction dynamics", *Pure Appl. Chem.* **1996**, *68*, 149-192.
- (148) Barberan-Santos M.N.; Martinho J.M.G. "The integration of kinetic rate equations by matrix methods", *J. Chem. Ed.* **1990**, *67*, 375-379.
- (149) Bandford, C. H.; Tipper, C. F. H. "Comprehensive Chemical Kinetics", Elsevier, Amsterdam, 1973.
- (150) Frost, A. A.; Pearson, R. G. "Kinetics and Mechanism", Wiley & sons, New York, 1961.
- (151) Gear G.W. "Numerical initial value problems in ordinary differential equations", Prentice-Hall, Englewood Cliffs, N.Y., (1987).
- (152) Weigert F.J. "A gear iterator" *Comput. Chem.* **1987**, *11(4)*, 273-280.

- (153) Gear C.W. "The automatic integration of ordinary differential equations", *Comm. ACM.* **1971**, *14*, 176-185.
- (154) Topping J. Errors of observation and their treatment, Chapman and Hall, London, 1978.
- (155) Cook R.D.; Weisberg S. Residuals and Influence in Regression, Chapman and Hall New York, 1982.
- (156) Gemperline P. "Practical Guide of Chemometrics", CRC Press (Taylor & Ferancis Group), USA (2006).
- (157) González-Hernández J.L. Tesis Doctoral, Departamento de Química Física, Universidad de Salamanca, **1978**.
- (158) Bisby R.H; Thomas E.W.K. "Kinetic Analysis by the Method of Nonlinear Least Squares" *J. Chem. Ed.* **1986**, *63 (11)*, 990-992.
- (159) Chau F.T; Mok K.W. "Multiwavelength analysis for a first-order consecutive reaction" *Comput. Chem.* **1992**, *16*, 239-242.
- (160) Bijlsma S; Louwerse D.J; Smilde A.K. "Estimating rate constants and pure UV-vis spectra of a two-step reaction using trilinear models" *J.Chemom.* **1999**, *13*, 311-329.
- (161) Chaudhary S; Van Horn J.D. "Biphasic kinetics in the reaction between amino acids or glutathione and the chromium acetate cluster, $[\text{Cr}_3\text{O}(\text{OAc})_6]^{+}$ " *Mutation Research* **2006**, *610*, 56-65.

Anexo



ELSEVIER

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Application the computational method KINMODEL(AGDC) to the simultaneous determination of kinetic and analytical parameters



M.M. Canedo Alonso*, J.L. González-Hernández, S. Encinar del Dedo

Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca, Pza. de los Caídos s/n, E-37008 Salamanca, Spain

ARTICLE INFO

Keywords:

Unconstrained optimization
 Computational program
 Computational chemical kinetics
 Kinetic modeling
 Resolution of mixtures

ABSTRACT

In a preliminary work we developed a computational method of kinetic treatment (KINMODEL(AGDC)) and studied the possibilities of its application for the determination of kinetic and non-kinetic parameters. In this new work, we studied the kinetic and/or analytical applicability of the KINMODEL(AGDC) program in a broad series of reactions of different complexities: (a) Simple reactions (reversible and irreversible), (b) Consecutive (reversible and irreversible), (c) Concurrent, (d) Competitive, (e) Cyclic. KINMODEL(AGDC) is a computational method that uses the AGDC mathematical optimization algorithm to determine different types of parameters. It is valid for the treatment of any reaction mechanism and allows the determination of different parameters from absorbance data. In all the reactions system included in the a, b, c, d and e groups, the KINMODEL(AGDC) methodology was applied for the determination of the initial concentrations of the species involved in the reaction mechanism, individually or jointly, with the kinetic constants of the elementary reactions comprising the mechanism. We analyze the influence of an extensive series of factors affecting the optimization process: the nature and number of the parameters to be optimized, the initial estimations of the parameters, the reaction mechanism and the relative values of the different rate constants of the mechanism considered. Since parameters of different natures and orders of magnitude are determined, we analyze the possibility of the existence of ambiguity in the solutions since it is common to find several (two or more) groups of parameters that fit the experimental kinetic data.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

In a preliminary work [1] we developed a computational method (KINMODEL(AGDC)) for the treatment of kinetic data, this program was used for the determination of kinetic and non-kinetic parameters, rate constants and molar absorption coefficients. In the present work we studied the kinetic and/or analytical applicability of the computational program and we determined different kinetic and analytical parameters by means of the treatment of kinetic absorbance data. The computational treatment of kinetic data in order to determine kinetic and/or analytical parameters and reaction mechanisms is of great importance in different fields of chemistry. The literature contains many references to works in which different computational treatments have been used to study reaction mechanisms, with both kinetic aims, for the determination of rate constants and the discrimination of reaction mechanisms, and for analytical purposes for the resolution of homogeneous

* Corresponding author.

E-mail addresses: mcanedo@usal.es (M.M. Canedo Alonso), jlgh93@usal.es (J.L. González-Hernández), sonsoles_e@usal.es (S. Encinar del Dedo).
 URL: <http://web.usal.es/jlgh93> (M.M. Canedo Alonso).

mixtures [2]. In all the various works, the authors use different mathematical algorithms and a broad set of software options has been developed for these algorithms to be implemented. Here we shall consider some of the most important ones and those most closely related to the present investigation.

Traditional methods of curve-fitting [3–10] allow the determination of kinetic, analytical and thermodynamic parameters through application of different mathematical algorithms implemented in computational programs by means of the treatment of spectrophotometric data. Other relevant methods of data treatment used for the determination of kinetic and/or analytical parameters, are the Classic Curve Resolution techniques and their modifications: Classical Curve Resolution-Hard modeling (CCR-HM) [11,12], Classical Curve Resolution-Soft modeling (CCR-SM) [13] or a combination of both, Classical Curve Resolution-Combining Hard-Soft modeling (CCR-CHSM) [14,15]. Other techniques used the so-called Kalman filter algorithm, in its extended version [16–18], which evaluates spectrophotometric data, allowing the prediction of concentrations and rate constants, and there are also methodologies based on Artificial Neural Networks (ANN) techniques [19–25]. In general, the methodologies for kinetic treatment can be found in different text books [26–28], which gather and describe both classic and current methods for data treatment and the determination of different parameters.

The KINMODEL(AGDC) computational method [1] allows the determination of different kinetic, non-kinetic and analytical parameters from kinetic absorbance data through application of the ADGC mathematical optimization algorithm [29–36]. This algorithm has been used in many computational methods for the treatment of different chemical systems and it turn has been used for the treatment of different types of kinetic data (potentiometric, spectrophotometric, . . .) for kinetic, analytical and thermodynamic purposes. Taking into account the results obtained with these methodologies, we have adapted the AGDC algorithm, developing a new version that allows the treatment of kinetic data of absorbance at one or several different wavelengths. The AGDC algorithm can be adapted easily for the processing of kinetic experimental data acquired with other techniques, introducing in the expression that allows to obtain the value of the sum of quadratic deviations (SSQ) the corresponding relation between the monitored variable and the concentration. The new version of the algorithm was implemented in the generalized KINMODEL(AGDC) kinetic computational program [1]. The program was used for the treatment of different system of reaction, determining the rate constants of the elementary stages comprising the mechanism and the molar absorption coefficients of the species involved and it was also applied for discrimination among reaction mechanisms. Currently, there are two versions of the program: one version in FORTRAN 90 for use in a DOS environment and another in C++, using JAVA applications, which allows it to be used in a WINDOWS environment. Both versions are based on a generalized numerical method that allows the treatment of any reaction mechanism and the determination of different parameters with different orders of magnitude.

The KINMODEL(AGDC) model is a numerical method for kinetic treatment that allows the determination of a series of kinetic, non-kinetic and analytical parameters: (1) the kinetic constants (k_r) of the “r” elementary reactions comprising the reactions system, (2) the molar coefficients of the “j” species at different working wavelengths ($\epsilon_{j,\lambda}$), (3) the kinetic constants (k_r) and molar absorption coefficients ($\epsilon_{j,\lambda}$) simultaneously, (4) the initial concentrations of the “j” species $[B_j]_0$ involved, (5) the kinetic constants (k_r) and initial concentrations ($[B_j]_0$) jointly and simultaneously via a single optimization process. In that preliminary work [1], we discussed the application of the KINMODEL(AGDC) program for the determination of the parameters indicated in points 1, 2 and 3; i.e., kinetic constants and molar absorption coefficients. Here we analyze the possibility of using the methodology for the determination of the parameters reflected in points 4 and 5; i.e., the initial concentrations of the species involved (point 4) and the initial concentrations and kinetic constants simultaneously (point 5). For these optimization processes (points 4 and 5) we studied the effect of a series of factors that affect the optimization process, such as the nature and number of parameters to be optimized, the value of the initial estimates of the parameters, the reaction mechanism and the relative values of the various rate constants of the mechanism considered. In the optimization process mentioned in point 5, we performed a joint determination of parameters of different types (initial concentrations and rate constants), such that we analyzed the possibility of the existence of several groups of parameters that fit the kinetic experimental data.

2. Theoretical and computational aspects

2.1. Theoretical aspects

The computational methodology KINMODEL(AGDC) [1] carries out the determination of the parameters whose values one wishes to know by means of the AGDC mathematical optimization algorithm [29–36], it is a second-order gradient method that minimizes the numerical function (SSQ) given by:

$$SSQ = \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} ((A_{i,\lambda})_C - (A_{i,\lambda})_E)^2, \quad (1)$$

where: N_d = Number of experimental data pairs; N_w = Number of working wavelengths; N_c = Number of chemical species; $(A_{i,\lambda})_E$ = Total absorbance value obtained at a wavelength λ ; $\epsilon_{j,\lambda}$ = Molar absorption coefficient of the species j at a wavelength λ ; $[B_j]_i$ = Molar concentration of the species j at time i ; $(A_{i,\lambda})_C$ = Total absorbance value calculated.

According to the Beer–Lambert law the total absorbance, $A_{i,\lambda}$, of a mixture formed by N_c chemical species ($l = 1 \text{ cm}$), will be:

$$A_{i,\lambda} = \sum_{j=1}^{N_c} A_{j,i,\lambda} = \sum_{j=1}^{N_c} \epsilon_{j,\lambda} [B_j]_i. \tag{2}$$

Therefore the SSQ function can be expressed as a function of $[B_j]_i$ according to:

$$SSQ = \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\sum_{j=1}^{N_c} (\epsilon_{j,\lambda} [B_j]_i) - (A_{i,\lambda})_E \right)^2. \tag{3}$$

To know $[B_j]_i$ it is necessary the establishment and subsequent resolution of the system of differential rate equations that reflect the variation of the concentration of each species B_j with time i . The variation in the concentration is given by the general differential equation [37,38]:

$$\frac{d[B_j]}{dt} = \sum_{r=1}^{N_r} \nu_{j,r} k_r \prod_{c=1}^{N_c} [B_c]^{\nu_{c,r}}, \tag{4}$$

where: k_r = rate constant of the reaction r ; $[B_c]$ = concentration of the species acting as reagents in the reaction r ($\nu_{c,r} < 0$); $\nu_{c,r}$ = stoichiometric coefficient of the species B_c in the reaction r .

The program has a subroutine (DIFFEQ) that numerically establishes, in an automatic and generalized manner, the set of differential rate equations for the kinetic model under consideration, according to the matrix of stoichiometric coefficients of the model $\nu_{j,r}$, the rate constants of the different reactions integrating the model k_r , and the initial concentrations of the different species involved in the model $[B_j]_0$. To determine the concentration of all the species in the period of time considered, $[B_j]_i$, the program performs the numerical resolution of the set of differential equations by the method of numerical integration based on the Gear algorithm [39], which is efficient and robust and which performs the integration of the system with great accuracy and affords good results even when the system is complex and has stiff characteristics [40].

The minimization of the SSQ function (Eq. (3)) was carried out by means of an iterative process that first uses as the movement vector ($\mathbf{p}^{(m)}$) the one indicated by the Gauss–Newton method [41,42]:

$$\mathbf{p}^{(m)} = -\mathbf{g}^{(m)} [\mathbf{H}^{(m)}]^{-1}, \tag{5}$$

where $\mathbf{g}^{(m)}$ and $[\mathbf{H}^{(m)}]^{-1}$ are respectively the gradient vector and the inverse of the Hessian matrix of the iteration m , whose terms are derived from the function to be minimized (SSQ) with respect to each of the parameters to be determined (\mathbf{X}). The identity of the components of the Gradient vectors and Hessian matrices is different depending on the parameters to optimize.

If the parameters to determine are the initial concentrations of the different species involved in the model $[B_j]_0$, then $\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$ are given by:

$$\mathbf{g} = 2 \begin{pmatrix} \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial[B_3]_0} \\ \vdots \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \end{pmatrix}, \tag{6}$$

$$\mathbf{H} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \right)^2 & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \right) & \dots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \right) \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \right) & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \right)^2 & \dots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \right) \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_3]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \right) & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_3]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \right) & \dots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_3]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \right) \\ \vdots & \vdots & \dots & \vdots \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_1]_0} \right) & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \frac{\partial(A_{i,\lambda})_C}{\partial[B_2]_0} \right) & \dots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial[B_{N_c}]_0} \right)^2 \end{bmatrix}. \tag{7}$$

If the parameters to determine are the kinetic constants (k_r) and initial concentrations ($[B_j]_0$) jointly and simultaneously from a single optimization process, then $\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$ are given by:

$$\mathbf{g} = 2 \begin{pmatrix} \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial k_1} \\ \vdots \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial k_r} \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \\ \vdots \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \text{RES}_{i,\lambda} \frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \end{pmatrix}, \quad (8)$$

$$\mathbf{H} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_1} \right)^2 & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_1} \frac{\partial(A_{i,\lambda})_C}{\partial k_r} \right) & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_1} \frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \right) & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_1} \frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \right) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_r} \frac{\partial(A_{i,\lambda})_C}{\partial k_1} \right) & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_r} \right)^2 & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_r} \frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \right) & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial k_r} \frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \right) \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \frac{\partial(A_{i,\lambda})_C}{\partial k_1} \right) & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \frac{\partial(A_{i,\lambda})_C}{\partial k_r} \right) & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \right)^2 & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \right) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \frac{\partial(A_{i,\lambda})_C}{\partial k_1} \right) & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \frac{\partial(A_{i,\lambda})_C}{\partial k_r} \right) & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \frac{\partial(A_{i,\lambda})_C}{\partial [B_1]_0} \right) & \cdots & \sum_{i=1}^{N_d} \sum_{\lambda=1}^{N_w} \left(\frac{\partial(A_{i,\lambda})_C}{\partial [B_{N_c}]_0} \right)^2 \end{bmatrix}. \quad (9)$$

The residuals RES are given by:

$$\text{RES}_{i,\lambda} = ((A_{i,\lambda})_C - (A_{i,\lambda})_E). \quad (10)$$

Calculation of the derivatives is performed numerically by means of the Central Differences method [39] and the inverse of the Hessian matrix by means of the Gauss elimination method [43] which determines an approximate inverse matrix, then its accuracy is improved through an iterative method of successive approximations [44].

The AGDC algorithm [29–36], performs a rigorous analysis and control of the movement vector and of each of its terms, and suitable modifications can be made if any errors are detected, thereby ensuring successful optimization. Once the optimization process has been achieved, the program determines the errors of the optimized parameters [45,46] and performs an exhaustive analysis of the residuals thus allowing the goodness of fit to be checked [47–49]. The statistical calculated parameters are the following ones: arithmetic mean, variance, standard deviation, square error, statistical measure of adjustment and Pearson function (χ^2).

The procedure followed by the methodology KINMODEL(AGDC) is indicated of schematic and reduced form:

1. Start the optimization process. $m = 0$
 - 1.1. Select the mechanism of reaction to be studied.
 - 1.2. Select the parameters to be optimized \mathbf{X} (\mathbf{k}_r and/or $[B_j]_0$, \mathbf{k}_r and/or $\varepsilon_{j,\lambda}$).
 - 1.3. Input data: Experimental data of absorbance/time $(A_{i,\lambda})_E$, convergence criteria CC, known parameter values, matrix of stoichiometric coefficients $\nu_{j,r}$ if 1.1.2. is selected, ..., Initial estimates of the unknown parameters $\mathbf{X}^{(0)}$.
2. Determine the $\text{SSQ}^{(0)}$ function (Eq. (3)).
 - 2.1. Calculate of concentrations of species $[B_j]_i$: Establish the rate differential equations system (Eq. (4)). Numerical solution of the rate differential equations system.
 - 2.2. Calculate the total absorbance at each wavelength $(A_{i,\lambda})_C$ (Eq. (2)).
3. AGDC optimization algorithm
 - 3.1. Calculate the vector of movement $\mathbf{p}^{(m)}$ ($\mathbf{p}^{(m)} = -(\mathbf{H}^{(m)})^{-1} \mathbf{g}^{(m)}$). Compute partial numerical derivatives of $(A_{i,\lambda})_C$ with respect to the parameters to be determined $\mathbf{X}^{(m)}$. Compute Gradient vector and Hessian Matrix ($\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$). Compute $(\mathbf{H}^{(m)})^{-1}$.
 - 3.2. Control and correction of the direction of the vector of movement $\mathbf{p}^{(m)}$ (AGDC).

- 3.2.1. If $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ go to 3.3.
- 3.2.2. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} < \varepsilon$ ($\varepsilon =$ scalar close to zero), $\mathbf{p}^{(m)+} = -\mathbf{g}^{(m)}$ and go to 3.3.
- 3.2.3. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} > 0$, $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$.
- 3.3. Control the length of the vector of movement $\mathbf{p}^{(m)}$ (AGDC).
 - 3.3.1. Compute the scalar ($\alpha^{(m)}$) by the method of Hartley [40].
 - 3.3.2. $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \alpha^{(m)} \mathbf{p}^{(m)}$
 - 3.3.3. Determinate the SSQ^(m+1) function (Eq. (3)).
 - 3.3.4. If the Goldstein–Armijo criterium [42] is satisfied go to 3.4.
 - 3.3.5. $\alpha^{(m)} = \alpha^{(m)}/2$ go to 3.3.2.
- 3.4. Calculate $\text{CON} = \left| \frac{\text{SSQ}^{(m+1)} - \text{SSQ}^{(m)}}{\text{SSQ}^{(m)}} \right|$
- 3.5. If convergence is not attained ($\text{CON} > \text{CC}$), set $m = m + 1$ and go to 3.1.
4. $\mathbf{X}^{(m+1)}$ = Optimized Parameters. Calculation of the errors of the parameters.
5. Statistical analysis of residuals.
6. END.

2.2. Computational aspects

There are two versions of the general program KINMODEL(AGDC), one version written in FORTRAN 90 language and compiled with the Microsoft Developer Studio (©Microsoft Corporation) compiler, for use in a DOS environment; and other version written in C++ language, using JAVA applications, which allows it to be implemented in a WINDOWS environment. Both versions are formed for a main program and a series of subroutines, in which the different treatments and calculations necessary for the optimization process of AGDC are carried out. The programs have been structured in the following parts:

1. Main program:

KINMODEL performs the following functions:

- Input data: reaction mechanism will be studied, parameters to be optimized and strategy, estimates of the unknown parameters whose value is to be determined, parameter values whose value is known, convergence criteria, experimental data absorbance/time, ...
- Automatically generates the model to study taking into account the data entered by the user.

2. Subprograms:

- 2.1. OPTIMAGDC, collection of subroutines that perform the optimization process of the different parameters by applying the AGDC algorithm: KINAGDC (k_r), CONCAGDC ($[\text{B}_j]_0$), KINCONAGDC (k_r and $[\text{B}_j]_0$), COEFAGDC ($\varepsilon_{j,\lambda}$) and SIMULAGDC (k_r and $\varepsilon_{j,\lambda}$).
- 2.2. GEARAGDC, a package of subroutines that generates (DIFFEQ) and solves (RESECDF) [50] the set of differential rate equations according to the model considered and therefore determines the concentration of each species in the time interval considered, $[\text{B}_j]_i$.
- 2.3. DERIVAGDC, a set of subroutines that calculate the numerical partial derivatives of the total absorbance of the sample with respect to the parameters to be optimized: DERCONC (with respect $[\text{B}_j]_0$), DERIVK (with respect to k_r) and DERIVCOEF (with respect to $\varepsilon_{j,\lambda}$).
- 2.4. INVERAGDC, calculates the determinant and performs the inversion of the Hessian matrix.
- 2.5. ESTADAGDC, subprogram that determines the errors of the parameters and performs statistical analysis of the residuals.

3. Results and discussion

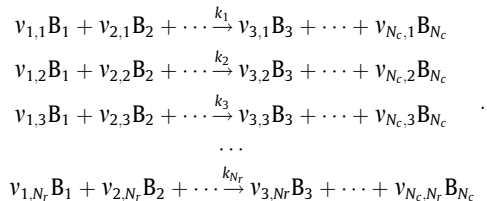
In the present work we studied the kinetic and/or analytical applicability of the KINMODEL(AGDC) program in a large series of reactions system. This computational methodology was used to perform two different optimization processes, on the one hand we studied the determination of the initial concentrations of the species involved in a given reaction mechanism and, on the other, we studied the jointly determination, in a single optimization process, of the initial concentrations and the kinetic constants of the different steps comprising the mechanism. In both cases we analyzed the influence of a series of factors that affect the optimization process: (a) the nature and number of the parameters to be optimized, (b) the initial parameter estimates, (c) the reaction mechanism and (d) the relative values of the different rate constants of the model considered.

We consider a chemical system in which exists N_c chemical species that can be involved in N_r chemical reactions, according to IUPAC rules [37,38] we can write the following system that it represents a generic system of reaction:

$$\begin{aligned}
 0 &= v_{1,1}B_1 + v_{2,1}B_2 + v_{3,1}B_3 + \dots + v_{N_c,1}B_{N_c} \\
 0 &= v_{1,2}B_1 + v_{2,2}B_2 + v_{3,2}B_3 + \dots + v_{N_c,2}B_{N_c} \\
 0 &= v_{1,3}B_1 + v_{2,3}B_2 + v_{3,3}B_3 + \dots + v_{N_c,3}B_{N_c} \quad \dots \\
 0 &= v_{1,N_r}B_1 + v_{2,N_r}B_2 + v_{3,N_r}B_3 + \dots + v_{N_c,N_r}B_{N_c},
 \end{aligned}$$

where: B_j = chemical species involved in the system of reactions; $r = (1, \dots, N_r)$, number of chemical reactions; $j = (1, \dots, N_c)$, number of i chemical species; k_r = kinetic rate constant of the r -th reaction; $v_{j,r}$ = stoichiometric coefficient of the species B_j in the r -th reaction, $v_{j,r} < 0$ when B_j plays only the role of reactant in the r -th reaction, $v_{j,r} > 0$ when B_j plays only the role of product in the r -th reaction.

Each of the reactions that compose the mechanism can be written also of the following way:



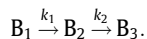
In this work, first we performed a broad study of the kinetic model formed by two first-order irreversible consecutive reactions ($B_1 \rightarrow B_2 \rightarrow B_3$), which is of great interest in chemical kinetics [51], in which we analyzed a series of situations that may arise when optimizing different types of parameters for this system. Then, we used the KINMODEL(AGDC) program for the treatment of kinetic data from other reaction systems: simple reactions (reversible and irreversible), consecutive reactions (reversible and irreversible), concurrent reactions, competitive reactions and cyclic systems.

3.1. Model $B_1 \rightarrow B_2 \rightarrow B_3$

The KINMODEL(AGDC) computational methodology was first applied to the kinetic model formed by two first-order irreversible consecutive reactions. For this system of reactions $N_c = 3$, $N_r = 2$ and the stoichiometric coefficient of the each specie “ j ” in each reaction “ r ” of the mechanism, are the followings ones:

$$\begin{aligned}
 r = 1 \quad v_{1,1} = -1 \quad v_{2,1} = 1 \quad v_{3,1} = 0 \\
 r = 2 \quad v_{1,2} = 0 \quad v_{2,2} = -1 \quad v_{3,2} = 1
 \end{aligned}$$

In a schematic way the system of reactions can be written as:



In this model it is carried out the determination of initial concentrations and the rate constants from the absorbance data generated at several wavelengths on which we imposed a noise similar to the experimental one. The total absorbance data ($A_{i,\lambda}$) were generated by the establishment and resolution of the set of differential rate equations and application of the Beer-Lambert law. For this mechanism, the total absorbance of the reacting mixture at each time “ i ” and for each wavelength, “ λ ”, is given by:

$$A_{i,\lambda} = \varepsilon_{B1,\lambda}[B_1]_i + \varepsilon_{B2,\lambda}[B_2]_i + \varepsilon_{B3,\lambda}[B_3]_i. \quad (11)$$

The data on total absorbance were generated considering the following values:

- Initial concentrations, $[B_1]_0 = 1.000 \times 10^{-3}$ M, $[B_2]_0 = [B_3]_0 = 0$ M
- The molar absorption coefficients of each species at each wavelength “ λ ”:
 λ_1) $\varepsilon_{B1,\lambda1} = 900 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$, $\varepsilon_{B2,\lambda2} = 600 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$, $\varepsilon_{B3,\lambda1} = 700 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$
 λ_2) $\varepsilon_{B1,\lambda2} = 970 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$, $\varepsilon_{B2,\lambda2} = 675 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$, $\varepsilon_{B3,\lambda2} = 678 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$
 λ_3) $\varepsilon_{B1,\lambda3} = 1000 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$, $\varepsilon_{B2,\lambda2} = 700 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$, $\varepsilon_{B3,\lambda3} = 900 \text{ mol}^{-1} \text{ cm}^{-1} \text{ dm}^3$

We then analyzed a series of factors that affect the optimization process, such as the nature, number and initial estimates of the parameters. We also analyzed the influence of the relative values of the rate constants by performing different experiments in which the relationship between both constants was modified. The values of the rate constants corresponding to the different experiments and the relationships between them ($R = k_1/k_2$) are as follows:

- (1) $R = 100$ ($k_1 = 0.500 \text{ min}^{-1}$ y $k_2 = 0.005 \text{ min}^{-1}$)
- (2) $R = 0.010$ ($k_1 = 0.005 \text{ min}^{-1}$ y $k_2 = 0.500 \text{ min}^{-1}$)
- (3) $R = 5.000$ ($k_1 = 0.500 \text{ min}^{-1}$ y $k_2 = 0.100 \text{ min}^{-1}$)
- (4) $R = 2.000$ ($k_1 = 0.100 \text{ min}^{-1}$ y $k_2 = 0.050 \text{ min}^{-1}$)
- (5) $R = 0.500$ ($k_1 = 0.050 \text{ min}^{-1}$ y $k_2 = 0.100 \text{ min}^{-1}$)

- (6) $R = 1.250$ ($k_1 = 0.500 \text{ min}^{-1}$ y $k_2 = 0.400 \text{ min}^{-1}$)
 (7) $R = 1.111$ ($k_1 = 0.500 \text{ min}^{-1}$ y $k_2 = 0.450 \text{ min}^{-1}$)
 (8) $R = 1.020$ ($k_1 = 0.500 \text{ min}^{-1}$ y $k_2 = 0.490 \text{ min}^{-1}$)
 (9) $R = 1.002$ ($k_1 = 0.500 \text{ min}^{-1}$ y $k_2 = 0.499 \text{ min}^{-1}$)
 (10) $R = 1.000$ ($k_1 = 0.100 \text{ min}^{-1}$ y $k_2 = 0.100 \text{ min}^{-1}$)

3.1.1. Determination of initial concentrations

We first applied the KINMODEL(AGDC) program for the determination of the initial concentration of the species involved in the model considered: $B_1 \rightarrow B_2 \rightarrow B_3$. Table 1 shows the results obtained when the following processes of optimization are realized: *experience 1*, the initial concentration of reagent $[B_1]_0$ optimized individually; *experience 2*, initial concentration of the reagent and of the intermediate species jointly, $[B_1]_0$ and $[B_2]_0$; *experience 3*, jointly the concentration of the three species, reagent ($[B_1]_0$), intermediate ($[B_2]_0$) and product ($[B_3]_0$). The table shows the values initially estimated for the different initial concentrations ($[B_j]_{0,EST}$) and the values of the same after applying KINMODEL(AGDC) ($[B_j]_0$), together with the value of the SSQ function obtained after the optimization process (SSQ_{FINAL}). The computational treatment was applied to a series of experiments in which the relationship between the rate constants was varied and in which different initial estimates of the concentrations were taken, considering all the situations: higher or lower values, distant or close to the real value. In all experiences, convergence was reached satisfactorily and the optimization process occurred correctly, although the values of the initial estimates departed from the real ones and hence this process was not affected either by the starting initial estimates or by the relationship between the rate constants. The values reached for sum of quadratic deviations (of the order of 10^{-7}) afforded a Standard Deviation of the fit in accordance with the noise imposed on the data. When joint optimization of the initial concentration of the reagent and the intermediate, B_1 and B_2 , (*experience 2*) or joint optimization of the initial concentration of the reagent, intermediate and product, B_1 , B_2 and B_3 , were carried out (*experience 3*), the value of the initial concentration of B_1 was determined correctly, obtaining a value that was used to generate data. Regarding the initial concentration of the other species (B_2 and B_3), it was observed that very low values were obtained (close to zero), which is logical because we started out only from B_1 . In some cases, negative values were obtained for the concentrations of B_2 and B_3 ; this is a consequence of the own numerical optimization process, which accumulates errors during the process, in many cases due to the “truncation” of the numerical values that are obtained in the different steps. Also, it is necessary to take into account that the optimization process performed had no restrictions.

3.1.2. Simultaneous and joint determination of initial concentrations and rate constants

Owing to the interest of the simultaneous determination of parameters of different nature and orders of magnitude, we applied the KINMODEL(AGDC) computational methodology to the determination of initial concentrations $[B_j]_0$ jointly and simultaneously with the rate constants (k_r) for the model of two first-order irreversible consecutive reactions from the data on total absorbance obtained at different wavelengths. As in the previous section, we studied the influence of the initial estimates and the relationship between the rate constants. Table 2 shows the results obtained for the simultaneous determination of k_1 and the initial concentration of the reagent, B_1 , (*experience 1*) and the simultaneous determination of k_1 , k_2 , and the

Table 1

Optimization of initial concentrations (Model $B_1 \rightarrow B_2 \rightarrow B_3$).

Exp.	k_1/k_2	100	0.010	5.000	2.000	0.500	1.250	1.111	1.020	1.002	1.000
1	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	1000.0	0.01	1000.0	0.1	1000.0	0.01	10000.	0.01	100.	10.0
	$[B_1]_0 \cdot 10^3 / \text{mol dm}^{-3}$	0.9999	0.9999	0.9999	0.9999	0.9999	1.0000	1.0000	1.0000	1.0000	0.9992
	$SSQ_{FINAL} \cdot 10^7$	3.6417	0.1055	0.6776	1.3824	0.2449	0.2646	0.2517	0.2241	0.2148	13.0393
2	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	0.1	0.1	0.01	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	$[B_2]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	0.001	0.01	0.01	0.0001	0.001	0.001	0.001	0.001	0.001	0.001
	$[B_1]_0 \cdot 10^3 / \text{mol dm}^{-3}$	0.9998	0.9999	0.9998	1.0000	1.0000	1.0000	1.0002	0.9999	0.9999	1.0000
3	$[B_2]_0 \cdot 10^{17} / \text{mol dm}^{-3}$	1.3332	0.2573	1.3299	-1.2958	-0.5297	-1.2904	63.957	0.3863	-0.7004	1.7637
	$SSQ_{FINAL} \cdot 10^7$	5.8274	1.7075	2.6932	2.6436	2.7813	2.7820	2.8460	2.9002	2.8362	1.0477
	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	10.0	100.0	10.0	10.0	1.0	10.0	1.0	1.0	10.0	10.0
	$[B_2]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	0.001	0.01	0.01	0.0001	0.001	0.001	0.001	0.001	0.001	0.001
	$[B_3]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
	$[B_1]_0 \cdot 10^3 / \text{mol dm}^{-3}$	1.0000	1.0001	1.0000	1.0000	0.9999	1.0002	1.0000	0.9999	1.0000	0.9999
3	$[B_2]_0 \cdot 10^{21} / \text{mol dm}^{-3}$	0.01	1.0000	0.0100	0.0100	0.0001	0.0100	0.0001	0.0100	0.1000	1.0000
	$[B_3]_0 \cdot 10^{27} / \text{mol dm}^{-3}$	0.1000	1.0000	1.0000	0.1000	0.0100	1.0000	0.0001	1.0000	0.1000	0.1000
	$SSQ_{FINAL} \cdot 10^8$	5.6279	8.4365	1.0986	0.8966	5.7812	3.9128	0.9987	1.0037	8.0051	2.9157

initial concentration of the reagent B_1 (*experience 2*). The table shows the initial estimates of the parameters to be optimized ($k_{r,EST}$ and $[B_1]_{0,EST}$), the relationship between the rate constants, the optimized values of the parameters (k_r and $[B_1]_0$) and the value of the SSQ function obtained once convergence had been reached. No problems were observed in the optimization process and in all the experiences parameter values were obtained that were very close to the real ones, together with acceptable values for SSQ, for the errors of the parameters and for the standard deviation of the fit (in accordance with the noise imposed on the data). Statistical analysis of residuals confirmed the goodness of fit in all experiences.

Joint determination of the rate constants and the initial concentration of the reagent may lead to a problem of ambiguity in the solution; i.e., there may be two or more groups of parameters that give two indiscernible minima of the SSQ function [18,51–53]. This problem of ambiguity appears on computing kinetic data corresponding to the intermediate species, B_2 ($[B_2]/\text{time}$ or absorbance of B_2/time). In this case, there are two solutions corresponding to the groups of parameters ($k_1, k_2, [B_1]_0$) and ($k'_1, k'_2, [B_1]'_0$) that afford two minima of the SSQ function that are equally valid from a mathematical point of view and are related to each other as follows:

$$k'_1 = k_2, \quad (12)$$

$$k'_2 = k_1, \quad (13)$$

$$[B_1]'_0 = (k_1/k_2)[B_1]_0. \quad (14)$$

Such ambiguity does not exist in the case of computing data on total absorbance, as may be seen from the results obtained in *experience 2*, shown in Table 2, always obtaining the same solution, ($k_1, k_2, [B_1]_0$), and hence a single minimum for the SSQ function, regardless of the starting estimates for the rate constants and the initial concentration of the reagent.

Taking into account the versatility and ease of application of the KINMODEL(AGDC) program, which allows the treatment of both data on total absorbance and absorbance data corresponding to any of the species involved in the system of reactions, we performed a series of experiments in which the optimization process was carried out by computing the absorbance data corresponding to the intermediate species B_2 . In this case, the value of absorbance at each time “ t ” and each wavelength “ λ ”, is given by:

$$A_{i,\lambda} = A_{2,i,\lambda} = \varepsilon_{B_2,\lambda}[B_2]_i \quad (15)$$

Table 2
Simultaneous optimization of individual rate constants and initial concentrations (Model $B_1 \rightarrow B_2 \rightarrow B_3$).

Exp.	k_1/k_2	100	0.010	5.000	2.000	0.500	1.250	1.111	1.020	1.002	1.000
1	$k_{1,EST} / \text{min}^{-1}$	1.0	0.0010	0.1000	0.0100	0.1	0.001	1.0000	0.001	1.0000	0.2
	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	1000.0	0.1000	10.00	10.00	0.1	0.1	10.00	100.00	100.00	0.0050
	k_1 / min^{-1}	0.5003	0.0050	0.5000	0.1000	0.0500	0.5000	0.5001	0.5000	0.5000	0.1006
	$[B_1]_0 \cdot 10^3 / \text{mol dm}^{-3}$	1.0000	1.0000	0.9999	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998
2	$SSQ_{FINAL} \cdot 10^8$	22.441	0.0639	5.4973	13.130	2.9336	2.5657	2.5512	2.3144	2.5617	19856
	$k_{1,EST} / \text{min}^{-1}$	1.0	0.003	0.5	100.0	0.1	0.1	0.4	1.0	1.0	0.2
	$k_{2,EST} / \text{min}^{-1}$	0.1	0.55	0.1	10.000	0.001	0.1	0.3	0.3	0.1	0.2
	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	100.00	5.00	10.00	1000.0	10.00	100.00	10.00	10.000	10.00	5.0
3	k_1 / min^{-1}	0.1000	0.0049	0.5001	0.1000	0.0501	0.4999	0.4999	0.5001	0.5002	0.1003
	k_2 / min^{-1}	0.0049	0.5492	0.0999	0.0499	0.0999	0.4004	0.4506	0.4888	0.4947	0.1009
	$[B_1]_0 \cdot 10^3 / \text{mol dm}^{-3}$	1.0000	0.9999	0.9999	1.0000	1.0000	0.9999	0.9994	1.0000	1.0000	0.9999
	$SSQ_{FINAL} \cdot 10^8$	8.4905	32.942	5.7196	8.4905	0.9226	23.586	21.466	20.279	156.95	198.56
4	$k_{1,EST} / \text{min}^{-1}$	0.1	0.0010	0.9	0.5	0.01	0.1	0.2	0.2	0.2	0.2
	$k_{2,EST} / \text{min}^{-1}$	0.001	0.1	0.2	0.01	0.5	0.09	0.1	0.1	0.1	0.2
	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	5.0	5.0	50.0	5.0	5.0	5.0	5.0	5.0	1.0	5.0
	$SSQ_{FINAL} \cdot 10^8$	20.235	0.7701	5.7404	9577.4	32.298	8.7957	2.5447	5.7925	86.579	15.632
4	$k_{1,EST} / \text{min}^{-1}$	0.1	0.1	0.2	0.01	0.5	0.3	0.1	0.1	0.1	0.2
	$k_{2,EST} / \text{min}^{-1}$	0.001	0.001	0.9	0.5	0.01	0.5	0.2	0.2	0.2	0.2
	$[B_1]_{0,EST} \cdot 10^3 / \text{mol dm}^{-3}$	500.0	500.0	5.0	5.0	5.0	1.5	5.0	5.0	1.0	5.0
	$SSQ_{FINAL} \cdot 10^8$	20.235	0.7701	5.7404	9577.4	32.298	8.7957	2.5447	5.7925	86.579	15.632
4	k_1 / min^{-1}	0.0050	0.4999	0.1000	0.0502	0.0997	0.4007	0.4543	0.4861	0.4765	0.1023
	k_2 / min^{-1}	0.4999	0.0050	0.4998	0.0995	0.0501	0.4989	0.4954	0.5046	0.5245	0.1010
	$[B_1]_0 \cdot 10^3$	100.0	0.00999	5.0000	1.9887	0.5008	1.2472	1.1100	1.0292	1.0508	1.0020
	$SSQ_{FINAL} \cdot 10^8$	20.235	0.7701	5.7404	9577.4	32.298	8.7957	2.5447	5.7925	86.579	15.632

Application of KINMODEL(AGDC) to absorbance data corresponding only to the intermediate species gave rise to two identical minima of the SSQ function corresponding to two different groups of parameters, as may be seen from *experiences* 3 and 4, shown in Table 2. Thus, it was observed that in the case in which joint determination of the rate constants and the initial concentration of reagent was carried out based on absorbance data corresponding to the intermediate species there were two different solutions (*experiences* 3 and 4), related to each other by Eqs. (12)–(14), affording a minimum value of the SSQ function that was identical in both solutions. These two solutions are not discernible from a mathematical point of view. In the case of the joint determination of these parameters it is recommendable to perform the data treatment on total absorbance or absorbance data corresponding to the reagent or final product, since in these cases there is no problem of ambiguity, as seen (*experience* 2).

3.1.3. Relationship $k_1/k_2 = 1$

The determination of initial concentrations individually and rate constants jointly with the initial concentrations by application of KINMODEL(AGDC) was performed in the extreme situation (although impossible from the chemical point of view) in which the rate constants are identical, $k_1 = k_2$; that is, $k_1/k_2 = 1$, in order to check whether the numerical treatment was able to discern between two identical kinetics. This aspect has been studied using other methodologies such as Neural Networks [19,21] for kinetic determinations of species, with rate relationships approximately equal to unity, although not quite so.

The limit mathematical situation of $k_1 = k_2$ is chemically unrealistic since it would mean that the chemical species B_2 would have the same reactivity as that of B_1 , leading to undetermination in the analytical solution of the differential rate equations (Eqs. (16)–(18)). For the $B_1 \rightarrow B_2 \rightarrow B_3$ reaction mechanism, the differential rate equations are as follows:

$$\frac{d[B_1]}{dt} = -k_1[B_1], \quad (16)$$

$$\frac{d[B_2]}{dt} = k_1[B_1] - k_2[B_2], \quad (17)$$

$$\frac{d[B_3]}{dt} = k_2[B_2]. \quad (18)$$

Assuming that initially ($t = 0$) there is only reagent B_1 ($[B_1]_0 \neq 0$, $[B_2]_0 = [B_3]_0 = 0$), the exact analytical solution of this system of differential equations provides the following expressions for the concentrations of the species B_1 , B_2 and B_3 :

$$[B_1] = [B_1]_0 e^{-k_1 t}, \quad (19)$$

$$[B_2] = \frac{k_1 [B_1]_0}{k_2 - k_1} (e^{-k_1 t} - e^{-k_2 t}), \quad (20)$$

$$[B_3] = [B_1]_0 - [B_1] - [B_2] = [B_1]_0 \left(1 - \frac{k_2}{k_2 - k_1} e^{-k_1 t} + \frac{k_1}{k_2 - k_1} e^{-k_2 t} \right). \quad (21)$$

For the case in which $k_1 = k_2$, the exact solution of the system of differential equations (Eqs. (19)–(21)) leads to a problem of undetermination in the calculation of the concentration of the intermediate species and the final product (B_2 and B_3), since the difference $k_1 - k_2 = 0$. Any method of optimization that uses the exact analytical solution to calculate the concentrations fails in this extreme case ($k_1 = k_2$). KINMODEL(AGDC) uses a numerical method to solve differential equations, such that this problem is avoided.

The last column of Tables 1 and 2, shows the results obtained with the KINMODEL(AGDC) program in the hypothetical case in which $k_1/k_2 = 1$ in the individual determination of the initial concentrations (Table 1) and the joint determination of the initial concentrations and rate constants (Table 2). The results show the success of the treatment in discriminating reactions in the limit situation when the rates are equal. No problems were found during the application of the KINMODEL(AGDC) program and it determined the parameters correctly. This is because the methodology of optimization along the treatment uses numerical methods, even for the resolution of the differential equations.

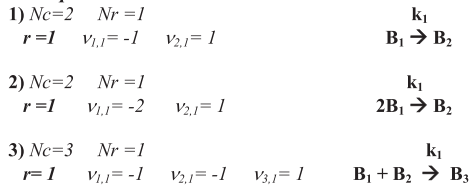
Other authors have analyzed the situation in which $k_1 = k_2 = 1.0 \text{ min}^{-1}$, $[B_1] = 1.0 \text{ M}$, $[B_2] = [B_3] = 0$ by applying methods that use models based on curve-solving techniques and modifications thereof (MCR-ALS) [14,18]. We applied the KINMODEL(AGDC) program to this situation, and like the procedures that use iterative approximate methods of gradient, KINMODEL(AGDC) correctly determined the parameters.

3.2. Determination of rate constants and initial concentrations in different systems of reactions

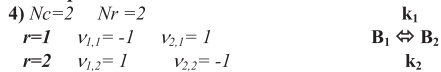
KINMODEL(AGDC) is a versatile and easy-to-use program that allows the treatment of any reaction mechanism by introducing the matrix of stoichiometric coefficients corresponding to the mechanism. Accordingly, we applied the program to different systems of reactions for the determination of the initial composition of a reactant mixture and the rate constants

of the different reactions occurring among these species, both individually and jointly. The KINMODEL(AGDC) program was used to study each of the models indicated below, which are those appearing most frequently in the literature:

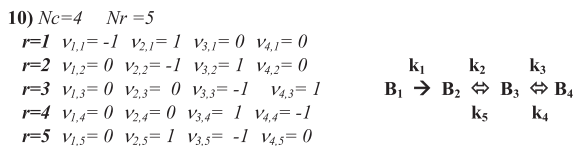
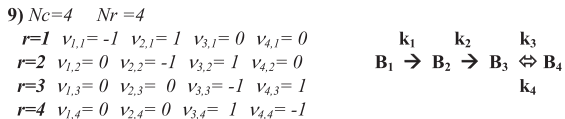
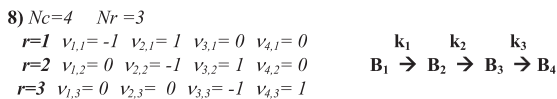
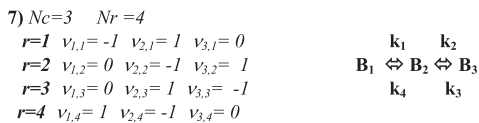
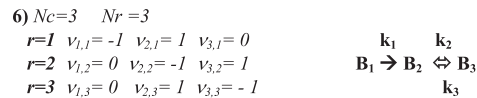
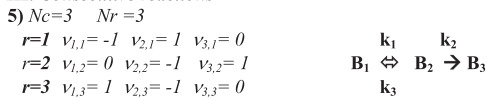
I. Simple irreversible reactions



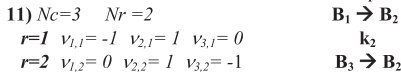
II. Simple reversible reactions



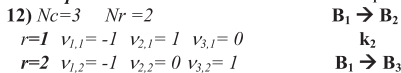
III. Consecutive reactions



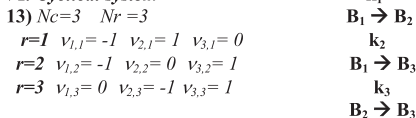
IV. Concurrent reactions



V. Competitive Reactions



VI. Cyclical system



For all models, we determined, both individually and jointly, the rate constants of the different steps comprising the mechanism and the initial concentrations of the different species involved in the system of reactions. Optimization of these parameters was performed from synthetic absorbance data (endowed with the appropriate noise), generated at three wavelengths, organizing the experiments according to a suitable prior design of experiments (DOE).

3.2.1. Determination of initial concentrations

KINMODEL(AGDC) was used for the individual determination of the initial concentrations ($[B_j]_0$) of the species involved in the above models when the values of the rate constants comprising the different steps of the system of reactions were known. Table 3 shows the results obtained upon treating the different systems, where the initial estimates for the initial concentrations, their optimized values and the value of the sum of SSQ obtained with the initial and optimized values. Determination of the initial concentrations of the different species involved in the system of reactions was performed successfully and the initial and final values of the SSQ function indicate the evolution of the optimization processes until a minimum value was reached. Good concordance is seen between the values obtained for the initial concentrations of the species acting as reagents and the values used to generate the data. Additionally, Table 3 shows the results obtained for the initial concentrations of the reaction products and/or intermediate compounds; their values are very close to zero (10^{-8} to 10^{-14}). This is logical, bearing in mind that the data we were working with were generated considering zero the initial concentrations of both the products and of the intermediate compounds. The values of the statistical parameters obtained, in agreement with the convergence conditions imposed, reveal the validity of the program for the determination of the composition of a mixture that changes with time, and the method can be proposed for the determination of the purity of unstable species.

3.2.2. Simultaneous and joint determination of initial concentrations and rate constants

Joint and simultaneous determination of the rate constants of the steps comprising the model (k_r) and the initial concentrations of the species involved in it ($[B_j]_0$) can be of great help when it is not possible to know the initial composition of the reagents and/or products and, further, the values of the constants are unknown. Accordingly, KINMODEL(AGDC) was applied for the joint and simultaneous determination of parameters of other types, such as k_r and $[B_j]_0$, from treatment of the data on total absorbance obtained at several wavelengths for the different mechanisms analyzed. Table 4 shows the results obtained for the simultaneous and joint determination of rate constants and reagent concentrations in the different reaction systems for models 1 to 6, 8, 11 to 13. Application of the program led to the determination of the correct parameters (those with which the data had been generated), detecting a decrease in the SSQ function, and its order of magnitude, in accordance with the error imposed on the simulated data. The values of the statistical parameters resulting from the Residuals Analysis show that the optimization process had been performed correctly. In models 7, 9 and 10, the same difficulties were encountered for the simultaneous determination of the four individual rate constants and the initial reagent concentration owing to different reasons, such as the increase in the number of parameters to be determined, their different orders of magnitude and their different nature. Table 4 shows the results obtained for mechanism 7, assuming that some of the parameters are known (k_1 and k_2), determining the other parameters (k_3, k_4 y $[B_1]_0$) jointly and simultaneously. With this strategy no difficulties are found in optimization in a single process where rate constants and initial concentrations are optimized. The results confirm the efficiency of the program for the simultaneous determination of rate constants and initial rates, a complex process owing to the different nature and order of magnitude of these parameters. This treatment reduces the experimental work to a considerable extent and is also of great use in situations where it is not possible to know the initial reaction conditions or there is some doubt about them.

From the results obtained from synthetic data endowed with a noise we can obtain the following conclusions: it is clear that when there is an increment of the number of species, the number of parameters to be optimized (k and the initial concentrations) increases. That is, the number of variables of the hyperspace of variation of the SSQ function increases and hence

Table 3
Initial concentrations optimized for different reaction systems (Models 1–13).

Model	$[B_1]_{0INIT}/$ mol dm ⁻³	$[B_2]_{0INIT}/$ mol dm ⁻³	$[B_3]_{0INIT}/$ mol dm ⁻³	$[B_4]_{0INIT}/$ mol dm ⁻³	SSQ _{INIT}	$[B_1]_0/$ mol dm ⁻³	$[B_2]_0/$ mol dm ⁻³	$[B_3]_0/$ mol dm ⁻³	$[B_4]_0/$ mol dm ⁻³	SSQ _{FINAL}
1	5.0E-3	0.00			18895.281	5.000E-4	4.023E-15			6.126E-9
2	1.0E-2	0.00			4224.381	0.999E-3	7.499E-7			1.402E-7
3	1.0E-3	1.0E-3			13.414	0.999E-3	1.000E-4			2.964E-7
4	1.0E-2	0.00			7320.937	0.999E-3	7.678E-11			8.165E-7
5	1.0E-2	0.00			5765.51	0.999E-3	3.374E-11			1.041E-7
6	1.0E0	0.00			6.892E7	0.999E-3	1.271E-11			1.027E-7
7	1.0E0	0.00			6.994E7	0.999E-3	4.033E-13			1.118E-7
8	5.0E-3	0.00	0.00	0.00	687.536	1.000E-3	-1.134E-9	1.535E-10	-3.937E-12	2.132E-8
9	5.0E-3	0.00	0.00	0.00	675.400	1.000E-3	-7.141E-10	2.011E-11	-1.426E-14	2.314E-8
10	5.0E-3	0.00	0.00	0.00	673.552	0.999E-3	7.282E-10	-2.737E-12	3.045E-12	1.219E-7
11	2.0E-3	0.00	2.0E-4		75.519	0.999E-3	1.264E-11	0.999E-4		0.955E-7
12	1.0E-10	0.00	0.00		84.041	1.000E-3	4.858E-9	4.966E-10		1.003E-7
13	1.0E-6	0.00	0.00		83.863	0.999E-3	9.949E-8	6.329E-8		0.983E-8

Table 4

Simultaneous optimization of individual rate constants and initial concentrations for different reaction systems.

Model	k_1/min^{-1}	k_2/min^{-1}	k_3/min^{-1}	k_4/min^{-1}	$[B_1]_0/\text{mol dm}^{-3}$	$[B_2]_0/\text{mol dm}^{-3}$	$[B_3]_0/\text{mol dm}^{-3}$	$[B_4]_0/\text{mol dm}^{-3}$	SSQ _{FINAL}
1	7.000E-4				5.000E-4				1.138E-10
2	1.000E0				0.999E-3				0.889E-7
3	2.003E-1				0.999E-3	0.999E-4			2.915E-7
4	1.000E-2	5.008E-3			0.999E-3				0.877E-7
5	0.999E-1	5.000E-2	0.997E-2		0.999E-3				0.897E-7
6	0.999E-1	4.999E-2	0.998E-2		0.999E-3				1.024E-7
7			1.000E-2	0.998E-2	0.999E-3				1.122E-7
8	1.000E-1	5.004E-2	2.997E-2		1.000E-3				5.3186E-9
11	0.999E-1	5.011E-2			0.999E-3		0.999E-4		0.935E-7
12	0.999E-2	5.000E-3			0.999E-3				2.237E-7
13	0.999E-2	5.000E-3	1.002E-4		0.999E-3				2.711E-7

the optimization process becomes much more complex which may lead to process failure. The model 7 is a paradigmatic example. As commented above, KINMODEL(AGDC) optimizes, with undoubted success, the four rate constants of model 7 and even the rate constants with one initial concentration (five parameters) without the need to know *a priori* the experimental values of the equilibrium constants K_1 and K_2 . KINMODEL(AGDC) is able to successfully optimize the five rate constants of models 9 and 10, which are much more complex and have a higher number of parameters than mechanism 7. On comparing these results with those reported in the literature, it is seen that the computational methods used there simultaneously and successfully optimize a maximum of three parameters (three rate constants) corresponding to models 5 and 6 [27]. The methods described in the literature are unable for the treatment of model 7 (four rate constants) since it is necessary to know *a priori* the experimental values of both equilibria constants that are relationships among the four rate constants ($K_1 = k_1/k_4$ and $K_2 = k_2/k_3$).

Acknowledgments

The authors gratefully acknowledge financial support to the “Universidad de Salamanca” to carry out this research in the frame of the 2 Investigation Projects USAL2005-B2-02 (2005) and USAL2008-B2-7 (2008).

References

- [1] M.M. Canedo, J.L. González Hernández, KINMODEL (AGDC): a multipurpose computational method for kinetic treatment, *J. Math. Chem.* 49 (2011) 163–184.
- [2] S.R. Crouch, A. Scheeline, E.S. Kirkor, Kinetic determinations and some kinetic aspects of analytical chemistry, *Anal. Chem.* 72 (2000) 53–70.
- [3] M. Maeder, A.D. Zuberbühler, Nonlinear least-squares fitting of multivariate absorption data, *Anal. Chem.* 62 (1990) 2220–2224.
- [4] S. Bijlsma, D.J. Louwerse, W. Windig, A.K. Smilde, Rapid estimation of rate constants using on-line SW-NIR and trilinear models, *Anal. Chim. Acta* 376 (1998) 339–335.
- [5] G. Huybrenchts, G. Van Assche, OPTKIN-mechanistics modeling by kinetic and thermodynamic parameter optimization, *Comput. Chem.* 22 (1998) 413–417.
- [6] F. Pérez Pla, J.F. Baeza Redón, R. Valero, A new algorithm for the kinetic data analysis, *Chemom. Intell. Lab. Syst.* 53 (2000) 1–19.
- [7] S. Bijlsma, H.F. Boelens, A.K. Smilde, Determination of rate constants in second-order kinetics using UV-Visible spectroscopy, *Appl. Spectrosc.* 55 (2001) 77–83.
- [8] S. Bijlsma, H.F. Boelens, H.C. Hoefsloot, A.K. Smilde, Constrained least squares methods for estimating reaction rate constants from spectroscopic data, *J. Chemom.* 16 (2002) 28–40.
- [9] B. Svir, O.V. Klymenko, M.S. Platz, KITFITSIM, a software to fit kinetic data to a user selected mechanism, *Comput. Chem.* 26 (2002) 379–386.
- [10] M. Maeder, Y.M. Neuhold, G. Puxty, P. Gemperline, Advances in the modelling and analysis of complex and industrial processes, *Chemom. Intell. Lab. Syst.* 82 (2006) 75–82.
- [11] G. Puxty, Y.M. Neuhold, M. Jecklin, M. Ehly, P. Gemperline, A. Nordon, D. Littlejohn, K. Basford, M. De Cecco, K. Hungerbühler, Multivariate kinetic hard-modelling of spectroscopic data: a comparison of the esterification of butanol by acetic anhydride on different scales and with different instruments, *Chem. Eng. Sci.* 63 (2008) 4800–4809.
- [12] N.C. Imlinger, C. Blattner, M. Krell, M.R. Buchmeiser, Hard-modeling of reaction kinetics by combining online spectroscopy and calorimetry, *J. Chemom.* 22 (2008) 758–767.
- [13] A. De Juan, E. Casassas, R. Tauler, Soft-modelling of analytical data, in: *Encyclopedia of Analytical Chemistry: Instrumentation and Applications*, Wiley, New York, 2000.
- [14] A. De Juan, M. Maeder, M. Martínez, R. Tauler, Combining hard- and soft-modelling to solve kinetic problems, *Chemom. Intell. Lab. Syst.* 54 (2000) 123–141.
- [15] J. Diewok, A. de Juan, M. Maeder, R. Tauler, B. Lendl, Application of a combination of hard and soft modeling for equilibrium systems to the quantitative analysis of pH-modulated mixture samples, *Anal. Chem.* 75 (3) (2003) 641–647.
- [16] R. Jimenez-Prieto, A. Velasco, M. Silva, D. Pérez-Bendito, Kalman filtering of data from first- and second-order kinetics, *Talanta* 40 (1993) 1731–1739.
- [17] B.M. Quencer, S.R. Crouch, Extended Kalman filter for multiwavelength, multicomponent kinetic determinations, *Analyst* 118 (1993) 695–701.
- [18] M. Gui, S.C. Rutan, Determination of initial concentration of an analyte by kinetic detection of intermediate product in consecutive first-order reactions used extended Kalman filter, *Anal. Chem.* 66 (1994) 1513–1519.
- [19] S. Ventura, M. Silva, D. Pérez-Bendito, C. Hervás, Artificial neural networks for estimation of kinetic analytical parameters, *Anal. Chem.* 67 (1995) 4458–4461.
- [20] D. Pérez-Bendito, M. Silva, Recent advances in kinetometrics, *Trends Anal. Chem.* 19 (6) (1996) 232–240.
- [21] S. Ventura, M. Silva, D. Pérez-Bendito, Estimation of parameters of kinetic compartmental models by use of computational neural networks, *J. Chem. Inf. Comput. Sci.* 37 (1997) 517–521.

- [22] C. Hervás, S. Ventura, M. Silva, D. Pérez-Bendito, Computational neural networks for resolving nonlinear multicomponent systems based on chemiluminescence methods, *J. Chem. Inf. Comput. Sci.* 38 (6) (1998) 1119–1124.
- [23] B. Kovacs, J. Tóth, Estimating reaction rates constants with neural networks, *Int. J. Appl. Math. Comput. Sci.* 4 (2007) 7–11.
- [24] N.H.T. Lemes, E. Borges, J.P. Braga, Rate constants and absorption coefficients from experimental data: an inversion procedure based on recursive neural networks, *Chemom. Intell. Lab. Syst.* 96 (2009) 84–87.
- [25] F. Amato, J.L. González Hernández, J. Havel, Artificial neural networks combined with experimental design: a “soft” approach for chemical kinetics, *Talanta* 93 (2012) 72–78.
- [26] B.G.M. Vandeginste, D.L. Massart, L.M.C. Buydens, S. Jong, P.J. Lewi, J. Smeyers-Verbeke, Data handling in science and technology, *Handbook of Chemometrics and Qualimetrics. Parts A and B*, vol. 20, Elsevier, Amsterdam, 1997.
- [27] P. Gemperline, *Practical Guide to Chemometrics*, CRC Press, USA, 2006.
- [28] M. Maeder, Y.M. Neuhold, *Practical Data Analysis in Chemistry*, Elsevier Science, Amsterdam, 2007.
- [29] M.N. Moreno, J.L. González-Hernández, M.A. Del Arco, J. Casado, Determination of macroscopic thermodynamic ionization constants under conditions of variable ionic strength by an optimization algorithm, *Comput. Chem.* 14 (1990) 165–168.
- [30] J.L. González-Hernández, M.M. Canedo, A. Domínguez-Gil, J.M. Lanao, Evaluation of thermodynamic ionization constants by computation of spectrophotometric and potentiometric experimental data obtained simultaneously from a flow system, *J. Pharm. Sci.* 81 (1992) 592–596.
- [31] A. Luján, J.L. González, M.M. Canedo, C. Grande, Determination of optimized individual kinetic constants of the aminolysis reaction in basic medium of β -Lactam antibiotics by numerical computational method, *J. Pharm. Sci.* 82 (1993) 1167–1171.
- [32] C. Grande, M.M. Canedo, J.L. González, KINAGDC-MW: a multipurpose program for the treatment of kinetic and equilibrium data, *Comput. Chem.* 20 (1996) 167–173.
- [33] J.L. González Hernández, M.M. Canedo, C. Grande, Optimization of kinetic parameters. Multipurpose KINAGDC(MW) non-linear regression program, *Chemom. Intell. Lab. Syst.* 39 (1997) 77–84.
- [34] M.M. Canedo, J.L. González Hernández, A new computational application of the AGDC algorithm for kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* 66 (2003) 63–78.
- [35] M.M. Canedo, J.L. González Hernández, ANALKIN(AGDC): a multipurpose computational program for the kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* 66 (2003) 93–97.
- [36] J.L. González, M.M. Canedo, C. Grande, Classic and multivariate modeling treatment of the kinetics and mechanism of isomerization of 5-cholesten-3-one catalyzed by sodium ethoxide, *Int. J. Chem. Kinet.* 38 (2006) 38–47.
- [37] K.J. Laidler, Symbolism and terminology in chemical kinetics, *Pure Appl. Chem.* 53 (1981) 753–770.
- [38] K.J. Laidler, A glossary of terms used in chemical kinetics, including reaction dynamics, *Pure Appl. Chem.* 68 (1996) 149–192.
- [39] C.F. Gerald, P.O. Wheatley, *Applied Numerical Analysis*, Adinon-Wesley, Massachusetts, 1984.
- [40] M.M. Canedo, J.L. González, F. Cuesta, J. Havel, Computation in kinetics V. A new KINDC computational method. Comparative study with other computer programs for kinetic treatment, *React. Kinet. Catal. Lett.* 48 (1) (1992) 77–83.
- [41] M.A. Wolfe, *Numerical Methods for Unconstrained Optimization*, Van Nostrand, Berkshire, 1978.
- [42] P. Gill, W. Murray, M.H. Wright, *Practical Optimization*, Academic Press Inc., London, 1981.
- [43] J.C. Mason, *Métodos Matriciales (Translated)*, Anaya Multimedia SA, Madrid, 1986.
- [44] B.P. Demidovich, I.A. Maron, *Cálculo Numérico fundamental (Translated)*, Paraninfo, Madrid, 1977.
- [45] K.J. Johnson, *Numerical Methods in Chemistry*, M. Dekker, New York, 1980.
- [46] J. Topping, *Errors of Observation and their Treatment*, Chapman and Hall, London, 1978.
- [47] R. Spiegel, *Estadística (Translated)*, McGraw Hill, Madrid, 1988.
- [48] R.D. Cook, S. Weisberg, *Residuals and Influence in Regression*, Chapman and Hall, New York, 1982.
- [49] M. Meloun, J. Havel, E. Höfeldt, *Computation on Solution Equilibria*, Ellis Hardwood, Chichester, 1988.
- [50] Harwell Subroutine Library Computer Science and System Division, AERE Harwell, Oxfordshire, 1984.
- [51] W.G. Jackson, J.M. Harrowfield, Consecutive, irreversible first-order reactions. Ambiguities and practical aspects of kinetic analyses, *Int. J. Chem. Kinet.* 9 (1977) 535–548.
- [52] I. Schechter, Error-compensated kinetic determinations by detecting the intermediate product in successive reactions, without prior knowledge of reaction constants, *Anal. Chem.* 63 (1991) 1303–1307.
- [53] J. Jaumot, P.J. Gemperline, A. Stang, Non-negativity constraints for elimination of multiple solutions in fitting of multivariate kinetic models to spectroscopic data, *J. Chemom.* 19 (2005) 97–106.

Combining artificial neural networks and experimental design to prediction of kinetic rate constants

J. L. González-Hernández · M. Mar Canedo ·
Sonsoles Encinar

Received: 29 January 2013 / Accepted: 19 March 2013 / Published online: 28 March 2013
© Springer Science+Business Media New York 2013

Abstract A “soft-modelling” computational approach of artificial neural networks (ANNs) combined with experimental design (ED) has been applied successfully in Chemical Kinetics for the prediction of kinetic rate constants. The system studied comprises two consecutive first-order reactions and the kinetic data were computed determining the values of both rate constants. The kinetic curves were distributed according to an ED, and the central star composite experimental design (CSCED) was chosen as the most appropriate. Computational treatments were performed on synthetic data endowed with noise, after which they were applied to the data measured in an experimental reaction between carbonyl cyanide 3-chlorophenylhydrazone with 2-mercaptoethanol, computing the experimental kinetic data of absorbance acquired at 3 wavelengths. The combined ANN and ED approach applied in chemical kinetics proved to be robust and of general applicability and has the advantage of being a “soft-modelling” method such that it was not necessary to solve the system of ordinary differential equations to determine the explicit mathematical function between the data and the kinetic rate constants. Additionally, upon using the CSCED experimental design, it was possible to substantially reduce the number of experiments.

Keywords Prediction kinetic constants · Artificial neural networks · Experimental design

J. L. González-Hernández (✉) · M. M. Canedo · S. Encinar
Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca,
37008 Salamanca, Spain
e-mail: jlgh93@usal.es

M. M. Canedo
e-mail: mcanedo@usal.es

S. Encinar
e-mail: sonsoles_e@usal.es

1 Introduction

Computational study of the kinetics of a reaction or a system of chemical reactions is mainly done for two reasons: to determine the kinetic and thermodynamic activation parameters in order to explain a plausible mechanism by which the reaction occurs, and, from an analytical point of view, to achieve a quantitative resolution of homogenous mixtures. The classic treatment used in such studies is the resolution of the system of ordinary differential equations (ODEs) to obtain an explicit mathematical function of the chemical-physical property being monitored with time (usually the absorbance), which includes the kinetic and thermodynamic parameters and initial concentration of the species. This function is different for each kinetic system formed by several species and different chemical reactions, and often involves not a few mathematical problems. Thus, serious problems arise as regards indistinguishability and /or non-unique identifiability, which lead to ambiguities in the solution of ODE system, as occurs in the case of a kinetic model with two consecutive first-order reactions, as studied here, where the function has 2 identical mathematical solutions that lead any curve-fitting method to fail. Even when the kinetic model is slightly more complicated (for example as from three species and four reactions), the ODE system lacks an exact mathematical solution and it is necessary to apply numerical treatments or, sometimes, approximate kinetic methods such as Steady State, Equilibrium State, etc. On other occasions, the function to be minimized has singular points (local minima, saddle points, etc.) that lead even the most robust of mathematical optimization methods to fail.

The computational treatment of the kinetic data for the determination of the kinetic, thermodynamic and analytical parameters is performed using (a) “hard-modelling,” approaches [1] where the systems can be exactly described by formulas, equations and the values of parameters, such as mathematical optimization techniques, curve-fitting techniques, Kalman filter algorithms etc. and (b) “soft-modelling” approaches when the exact description is not known or too complex, as multivariate curve resolution (MCR-ALS) [2], where kinetic data relating to absorbance acquired at multi-wavelengths are computed at different times. It is also possible to apply methods that include combinations of parts of both, giving rise to a hybrid approach, as is the case of MCR-PLS [3]. The “soft-” approaches have certain advantages since they act without prior knowledge of the model, such that it is not necessary to solve the ODE system; with this one avoids the mathematical problems of the explicit function mentioned for “hard-” approaches. However, these latter tend to be more robust and are useful and efficient when the kinetic model is known and the function resulting from the resolution of the ODE system is known, is simple, and does not have the afore-mentioned mathematical problems.

In the literature addressing computational kinetic treatment, a series of earlier works [4,5] is of interest where the authors applied different methods of mathematical optimization for the determination of kinetic and thermodynamic activation parameters. These works were the precursors of later ones in which the authors developed different computational programs for application in kinetic, analytical and chemical equilibrium studies KILET [6,7]. To avoid the difficulties found in classic optimization methods, at our laboratory we have designed a new and robust algorithm that has been implemented

in different computational programs (KINAGDC, ANALKIN(AGDC)[®], KINMODEL (AGDC),...etc.) developed for the treatment of equilibria [8] and the kinetic study of chemical systems [9–11]. These programs were applied to a broad range of reaction models, successfully determining rate constants, analyte concentrations in multicomponent mixtures (static and dynamic), molar absorption coefficients, etc. Additionally, they allowed us to discriminate between reaction mechanisms, deciding which was the most likely one and the one that best fitted the kinetic data. Later, we applied artificial neural network (ANN) techniques by the ANN application of Matlab [12] for the determination of kinetic parameters. A new approach (ED-ANN) was adapted to the treatment of kinetic concentration data [13], grouped in a particular input matrix (“training” matrix), whose rows jointly contained a sequential ordering of the data and parameters and whose computation was performed with the Simulator Trajan Neural Network and STATISTICA V.6. ANNs treatment of kinetic data has been used in multicomponent kinetic determinations, for analytical purposes [14, 15], computing the kinetic data acquired after the reaction of a reagent with the components of the mixture. Moreover, ANN techniques have been used to study the behaviour of chemical reactors in the field of Chemical Engineering [16–18] and the application in Kinetics of an inversion procedure based on recursive neural networks [19].

In the present work we applied an approach combining methods of ANN and experimental design (ED) for the prediction of kinetic rate constants of a system of 2 consecutive first-order reactions. This was a “soft-modelling” approach consisting of application of the trained ANN with the optimum architecture and topology resulting from the “training” process, able to determine the kinetic parameters without any *a priori* knowledge of the explicit mathematical function that was the solution of the ODE system correlating the data and the kinetic rate constants. The approach was also applied to the measured experimental kinetic data obtained from an experimental reaction taking place in the laboratory.

2 Theoretical aspects

2.1 Chemical kinetic aspects

Let us consider a chemical system formed by n_r chemical elementary (or concerted) reactions where n_s chemical species can be involved. According to IUPAC's norms [20], the system of reaction can be represented:

$$\begin{aligned}
 0 &= \nu_{1,1}B_1 + \nu_{2,1}B_2 + \nu_{3,1}B_3 + \cdots + \nu_{n_s,1}B_{n_s} \\
 0 &= \nu_{1,2}B_1 + \nu_{2,2}B_2 + \nu_{3,2}B_3 + \cdots + \nu_{n_s,2}B_{n_s} \\
 0 &= \nu_{1,3}B_1 + \nu_{2,3}B_2 + \nu_{3,3}B_3 + \cdots + \nu_{n_s,3}B_{n_s} \\
 &\dots\dots\dots \\
 0 &= \nu_{1,n_r}B_1 + \nu_{2,n_r}B_2 + \nu_{3,n_r}B_3 + \cdots + \nu_{n_s,n_r}B_{n_s}
 \end{aligned}$$

it can be expressed for the r -th chemical reaction with the generic equation,

$$0 = \sum_{j=1}^{n_s} \nu_{j,r} B_j \tag{1}$$

where

- B_j = chemical species involved in the system of reactions
- $r = (1, \dots, n_r)$ number of chemical reactions
- $j = (1, \dots, n_s)$ number of chemical species
- k_r = kinetic rate constant of the r -th reaction
- $\nu_{j,r}$ = stoichiometric coefficient of the species B_j in the r -th reaction
- $\nu_{j,r} < 0$ when B_j plays only the role of reactant in the r -th reaction
- $\nu_{j,r} > 0$ when B_j plays only the role of product in the r -th reaction

When the reaction is an elementary or concerted one, the absolute values of the kinetic order ($z_{1,r}$) and stoichiometric coefficient of B_j coincide, that is $|\nu_{1,r}| = |z_{1,r}|$. The rate differential equation of the chemical species B_j in the r -th is given by

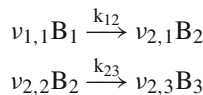
$$\frac{d[B_j]}{dt} = k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \tag{2}$$

where B_l are the species playing only the role of reactants in the r -th reaction ($\nu_{l,r} < 0$). Each chemical species can take part in several reactions and the rate differential equations will be the sum extended over those reactions where the reactant B_l appears, obtaining a system of ODEs according to the generic equation,

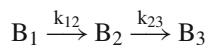
$$\frac{d[B_j]}{dt} = \sum_{r=1}^{n_r} k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \tag{3}$$

The general solution of the system of rate ODEs give the explicit function of the concentrations of the all species with the time ($[B_j]_{t_i}$).

The chemical system formed by 2 first order consecutive reactions has been studied in the present work and can be represented as



Considering $\nu_{1,1} = \nu_{2,2} = -1$ and $\nu_{2,1} = \nu_{2,3} = 1$, we have



According this, the system of ODE can be expressed using matrix notation [21] as

$$\frac{d}{dt} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12} & 0 & 0 \\ k_{12} & -k_{23} & 0 \\ 0 & k_{23} & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \tag{4}$$

When $[B_1]_0 \neq 0$ and $[B_2]_0 = [B_3]_0 = 0$, the integration of the ODE gives the following expressions of the concentrations for B_1 , B_2 and B_3

$$\begin{aligned} [B_1] &= [B_1]_0 e^{-k_{12}t} \\ [B_2] &= \frac{[B_1]_0 k_{12}}{k_{23} - k_{12}} \left(e^{-k_{12}t} - e^{-k_{23}t} \right) \\ [B_3] &= [B_1]_0 - [B_1] - [B_2] = [B_1]_0 \left(1 - \frac{k_{23}}{k_{23} - k_{12}} e^{-k_{12}t} + \frac{k_{12}}{k_{23} - k_{12}} e^{-k_{23}t} \right) \end{aligned} \quad (5)$$

According with the Lambert-Beer-Bouguer law, the expression of the absorbance measured, will be:

$$A_{j,t_i}^\lambda = \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad (6)$$

where A_{j,t_i}^λ is the absorbance of the species B_j at the, time t_i and path length 1 cm and ε_j^λ is the molar absorption coefficient of B_j at the wavelength λ . The absorbance of the mixture (A_{T,t_i}^λ) can be expressed as:

$$A_{T,t_i}^\lambda = \sum_{j=1}^{n_s} A_{j,t_i}^\lambda = \sum_{j=1}^{n_s} \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad (7)$$

2.2 Artificial neural networks

Artificial neural networks (ANNs) are parallel interconnected networks of simple computational elements called neurons and are structured in layers that are intended to interact with the objects of the real world in a similar way to the biological nervous systems [22]. Parallel processing is the ability of the brain to simultaneously process incoming stimuli of differing quality. The multilayer neural network uses sets of input data and parameters (called “targets”), distributed in 2 input matrices when Matlab is applied. The elements of the input matrix are the data, where one row contains a single curve and all the curves thus obtained (n_c) are grouped in an “input data” matrix. The “target matrix” is formed by the sets of parameters (n_p). In our case, the input data matrix contained the kinetic data of all curves expressed in A_T , A_j , $[B_j]$ or α_j (molar fraction of the species B_j) and the “target matrix” ($n_c \times n_p$) contained the set of kinetic rate constants (k_{mn}). Formally, a multilayer neural network is an oriented graph in which the nodes represent a set of processing units, called neurons, and the connections represent the information flow channels. Each connection between two neurons has an associated value called “weight” which specifies the strength of the connection between neurons. Positive and negative values determine excitatory and inhibitory connections, respectively. The choice of a specific class of networks for the approximation of a nonlinear map depends on a variety of *factors* dictated by the context and is related to the desired accuracy and the prior information available concerning the input-output pairs.

The first layer of a multilayer neural network contains neurons that receive the input data values from the elements of the input data matrices. This information is transmitted from the i -th neuron of a layer to the j -th neuron of the subsequent one, with a weight w_{ji} . A neuron parameter (“bias”) is summed with the weighted inputs of the neurons and passed through the transfer function to generate the output of the neurons.

The layer following the input one is called “hidden”. In each neuron of a “hidden” layer the weighed inputs coming from the previous one are summed with each other and added to a “bias”. The result is then transformed by means of a suitable mathematical function to obtain an output called “activation of the neuron”, which is transferred to the neurons in the next layer after another weighing step. The output parameters values are calculated in the last layer (“output” layer) by means of a suitable transformation function.

The process described is called to as the “training” of the multilayer neural network and constitutes an iterative method where after each iteration (“epochs”), the calculated values of the parameters are grouped in the “output matrix” (b_{ij}^{output}) and they are compared with those of the corresponding curve in the “target matrix” (b_{ij}^{target}) as shown the Scheme 1.

The value of the mean squared error (MSE), expressed in absolute value, is calculated according expression:

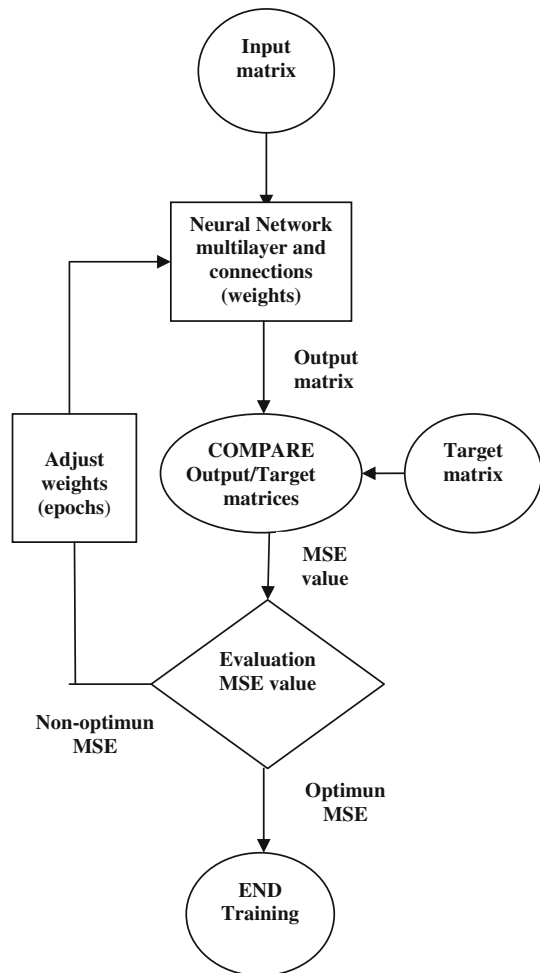
$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{output} - b_{ij}^{target})^2}{n_p \cdot n_c} \right)^{1/2} \quad (8)$$

where n_c is the number of curves and n_p is the number of parameters and $n_c \times n_p$ are the dimensions of both matrices (“output matrix” and “target matrix”).

During the process of “training”, “weights” and “bias” values are modified with suitable mathematical optimization algorithms in order to minimize the calculated values of MSE in each *epoch*. In the present work, the *back-propagation* algorithm was used. The iterative process finishes when the minimum value of MSE is reached, after which the “training” process can be considered to be completed.

It is necessary to know the optimal architecture and topology of the multilayer neural network in order to obtain the best results when ANN is applied to the system under study. This can be performed using the methods described in a recent review [23]. We have used a method of “trial and error” by minimizing the MSE values obtained for the different possible configurations of the same number of “hidden” layer/s. It must to determine the minimum value (optimum) of the MSE for all possible configurations for the “hidden” layer/s chosen. For each “hidden” layer, a graph of MSE values vs. the number of neurons shows that initially, for the lower configuration, the value of the MSE decreases rapidly when the number of neurons increases, but after a constant value or a poor improvement is obtained. The optimum number of neurons in that “hidden” layer is given by the point of intersection of the two branches of the graph. Sometimes, a small minimum appears near this intersection point. The architecture of the neural network can be written in abbreviated notation as (n_{inp} , n_{hid} , n_{out}), where

Scheme 1 Flowchart of the iterative method of the optimization of output/target values in the “training” of the neural network



n_{inp} is the number of neurons in the “input” layer, n_{hid} in the “hidden” layer and n_{out} in the “output” layer.

Neural network “training” is completed with the processes of “validation” and “testing”. These are 2 control and verification processes of the iterative minimization method between the elements of the “output” and “target” matrices. Among the different curves comprising the “input” matrix, random choice is made of a percentage of the total, established previously (5 %, 10 % . . .), which gives rise to a “sub-matrix” of input curves that are subjected to iterative optimization until a minimum MSE value is reached. It is thus possible to verify the validity of the “training” process by ensuring that it is convergent, that it has an appropriate termination, and that there not been any overfitting, since any possible “overtraining” has taken been into account. Validation is completed when in a given number (≥ 6) of consecutive “epochs” the MSE remains constant or shows a slight tendency to increase. The “testing” process is similar, except

that the control of the process is performed by controlling the computation time instead of the number of “epochs”.

The process of “prediction” consists of the determination of the unknown parameters from a set of experimental data after application of the optimal and trained neural network. Obviously, the elements of the “target” matrix are unknown for this “prediction” process, and only the “input data” matrix is provided to the neural network. In our case, the elements of the “input data” matrix in the process of “prediction” are experimental kinetic values (A_T , A_j , $[B_j]$, α_j , etc), acquired from a system of reactions developed at the laboratory.

2.3 Experimental design

The experimental design (ED) is a method to select within the working space according to a well-distribution in order to extract the maximum amount of information. The implementation of a suitable ED is crucial to ensure the success of the “training” process of neural networks. In our case, in the ED two *factors* were involved (k_{12} and k_{23}), whose “responses” are the kinetic data of the base of the input curves, which can be of 2 types according to whether they correspond to the contribution of all the species (A_T) or to the individual contribution of a single species (A_j , $[B_j]$ and α_j). In the ED of the neural network it is necessary to consider 2 variables: the limit values of both *factors* that configure the “experimental domain” and their relative values (k_{12}/k_{23}). Both variables must ensure that the binary combinations of the kinetic constants will generate a set of kinetic curves that will have sufficient information to ensure an optimal “training” process of the neural network. Within the interval of time considered, 2 requisites must be reached: (a) the kinetics must achieve a conversion of at least of 80 % ($\xi'_{\text{end}} = 0.8\xi'_{\text{max}}$) and (b) the values of the kinetic data must range within at least 40–50 % so that the neural network will be able to discriminate clearly between the different input curves. Additionally, the number of “levels” of the *factors* of the ED must be suitable if one is to avoid useless computational work and avoid large differences in the spacing of the values of the “responses”. Accordingly, to optimize the ANN “training” process the kinetic curves of the “input” matrix must have efficient kinetic information and must be correctly distributed according to the choice of a suitable experimental design and an appropriate “experimental domain”. Bearing in mind the characteristics of the kinetic system under study, we have chosen as the most suitable one, the ED corresponding to the *central star composite experimental design* (CSCED) distribution [24] for the combining with the ANNs technique.

3 Computational aspects

The computational work has been performed using Matlab divided in 3 different aspects: (a) the general computational treatment of ANN by means of the application of Matlab (“Neural Networks Toolbox”) with the creation of user’s interfaces (*GUI*) including the appropriate analysis of Residuals and errors (MSE, SD, etc) (b) design and performing by us of specific computational executable programs (##.m type) in the Matlab environment using “M” language, functions and applications for obtaining

synthetic data, mathematic optimization, fitting functions, etc. and (c) complex 3D graphic representations, endowed of motion and looking for the best perspective in order to obtain an easy visualization of the figure. In the case of the simpler plots other software was used (Origin 8.6 and QTplot)

4 Results and discussion

4.1 Organization of the kinetic data

We considered several types of organizations of the kinetic data in the input matrices to enrich the input kinetic information that would guarantee a more efficient “training” process. The neural network should be endowed with the highest number possible of input options for organizing the data, and this must lead to concordant results. 3-D plotting (A_T /number of curves/time) of the curves corresponding to the A_T data can be seen in Fig. 1a. In the curves of the individual kinetic variables (A_j , $[B_j]$ and α_j) we performed a sequential organization; that is, first the data set corresponding to the species B_1 , then those of the species B_2 , and finally those of the species B_3 (Fig. 1b). An especially applicable ordering for the kinetic system studied here (3 species and 3 coordinate axes) can be seen in a 3D plot (of the “banana” type) upon representing each individual kinetic variable directly on each of the 3 coordinate axes where all points belong to the same plane (Fig. 1c).

4.2 Training of the ANN. optimal architecture

In order to determine the optimum architecture of the neural network we evaluated the influence of the different variables on the “training” process of the ANN, assessing the effect of each of them in each layer.

4.2.1 “Input” layer

To determine the optimum number of curves we analyzed the ideal number of kinetic data of each kinetic curve. We tested 25, 50, 75 and 100 values, 50 data (50 neurons) proving to be optimum, since the improvement in the results obtained with 75 and 100 values did not justify the huge increase in computation time involved. The ideal maximum range of variation of both *factors* was the same for both, with limit values of 3.12×10^{-2} and $1.19 \times 10^{-1} \text{ min}^{-1}$. We varied the number of *levels* and *nodes* of both *factors*, analyzing the responses made. We checked the configurations formed by 29, 37, 45 and 53 nodes or curves, being the optimum configuration proved to be that formed by 45 nodes corresponding to 9 levels and 4 *sub*-levels (Fig. 2). This value offered a sufficient number of *responses*, with no large differences in the spacing, and also is a suitable number of curves for the computational work to be carried out. The dimensions of the *input* matrix were 50×45 .

Since the kinetic curves were generated synthetically, the percentages of curves in the “validation” and “testing” processes may be high with respect to the “training” curves. We tested several values, percentages of 80/10/10 proving respectively to be

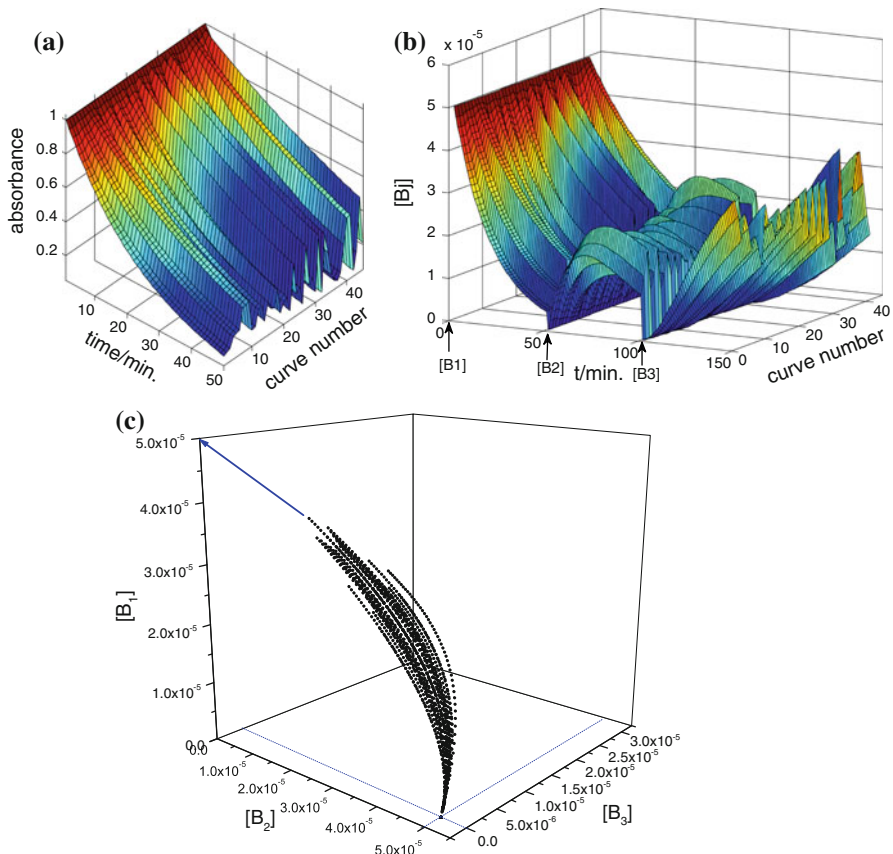


Fig 1 3D plots of the 45 kinetic curves taking part in the curve base of the “input matrix” used for the “training” process of the neural network, **a** each curve contains 50 data of synthetic data of the total absorbance (A_T) endowed with “noise”; **b** 150 data each curve of synthetic data of the total absorbance (A_T) endowed with “noise” with a sequential organization **c** “banana” type plot after representation of the concentrations of the 3 species $[B_1]$, $[B_2]$ and $[B_3]$ on each one of the 3 coordinate axes

optimum for “training”, “validation” and “testing”, since these percentages ensure a sufficient number of curves for the “training” process. A *random error (noise)* must be imposed on the synthetic data, with a magnitude similar to the intrinsic experimental error of the measurement; that is, $\pm 1 \times 10^{-4}$ to the absorbance data and $\pm 1.0 \times 10^{-8}$ to those of concentration ($[B_1]_0 = 5.50 \times 10^{-5} \text{ mol dm}^{-3}$).

4.2.2 “Hidden” layer/s

Determination of the optimum number of “hidden” layers and that of the neurons of each of them is crucial for the determination of the optimum architecture of the neural network, such that it is necessary to test the highest number of cases possible. The procedure, of the *trial and error* type, consist of initially fixing the number of “hidden” layer/s, carrying out the “training” of the neural network with different configurations,

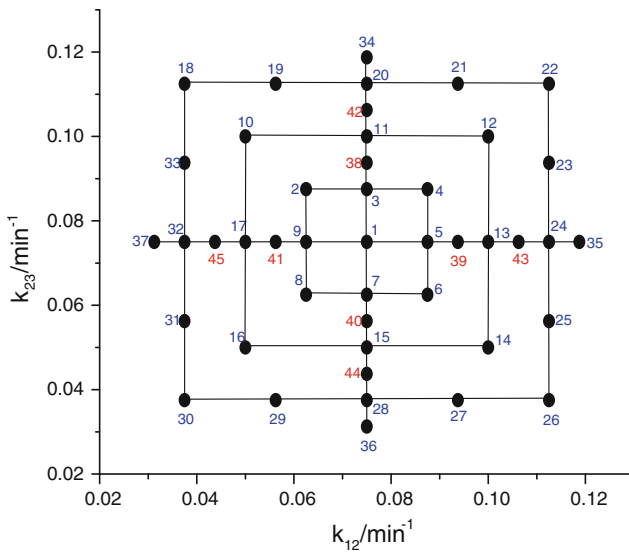


Fig 2 Experimental design (CSCED) constituted by 2 factors (k_{12} and k_{23}) and 45 points distributed in 9 levels (points 1–37) and 4 sub-levels: (points 38–45) taking part in the curve base of the “input matrix” used for the “training” process of the neural network

determining the values of MSE, and plotting the values of the MSE values versus the number of neurons for each configuration. As explained above, the observed minimum or the point of intersection of the two branches of the plot, provided (on the abscissa) the optimal number of neurons in that “hidden” layer.

For the A_T input data and a structure of a single “hidden” layer, we obtained an optimal configuration of 5 nodes (Fig. 3a), and 6/10 as the optimal configurations in the case of 2 “hidden” layers (Table 1), obtaining with this latter multilayer structure better results than that of a single hidden layer. In the case of the individual input data (A_j , $[B_j]$ and α_j) and a single “hidden” layer, a configuration of 8 nodes was obtained as the optimum value (Fig. 3b).

4.2.3 “Output” layer

The number of neurons of the “output” layer corresponds to the number of parameters to be predicted; in our case 2. Since Matlab offers the possibility of the application of several optimization algorithms, we tested all of them, choosing as the best the one that provided the best results. We then evaluated the results obtained upon applying the Steepest Descent, Fletcher-Reeves, Scales Conjugate Gradients, Quasi-Newton, BFGS Algorithm and Levenberg-Marquardt (LM), etc. algorithms, the latter (LM) being the one finally chosen since its afforded acceptable MSE values within a reasonable computing time and a suitable number of “epochs”.

According with the above we performed the ANN “training” process in the computation of synthetic data endowed with *noise* with the neural network on the basis of the optimal architecture that we obtained, with the following structure: the *input* layer

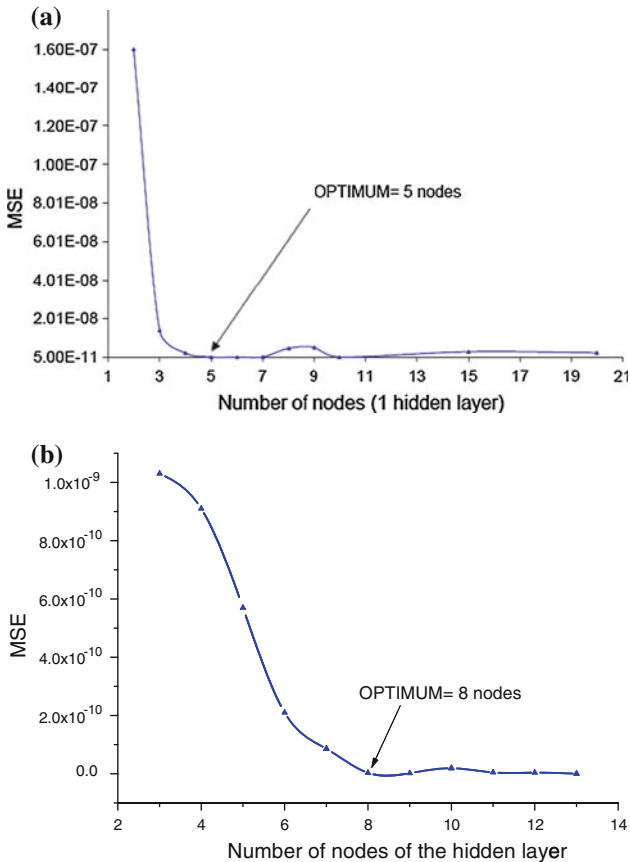


Fig 3 Optimization of the number of neurons (nodes) of the single “hidden” layer, **a** from the computation of individual input data (A_j , $[B_j]$ and α_j) and **b** when A_T input data are computed

and the “output” layer had 50 and 2 neurons, respectively, while the composition of the hidden layer depended on the type of kinetic data of the base of the curves. In the case of individual data (A_j , $[B_j]$ and α_j), the optimum architecture was (50, 1, 2) with a single “hidden” layer, whose suitable configuration had 5 nodes. For the A_T data, the optimal architecture was (50, 2, 2) with 2 hidden layers and the configuration of each layer being of 6/10 according with the values of Table 1.

The process of “training” the optimum neural network was performed with the following characteristics; (a) the kinetic data of the “input” matrix formed a base of curves composed of 45 curves, in agreement with a CSCED chosen, with 50 data for each curve and a standard matrix organization in the case of the A_T data and a sequential one for the A_j , $[B_j]$ and α_j data. (b) the noise value imposed on the synthetic data $\pm 1 \times 10^{-4}$ for the absorbance data and $\pm 1.0 \times 10^{-8}$ for the concentration data; (c) in all the curves we considered kinetic data with a maximum conversion of no less than 82% ($\xi'_{\text{end}} = 0.82\xi'_{\text{max}}$) and the differences between their limit values were always higher than a value of around 50%; (d) the values of the percentages of

Table 1 Values of MSE for the “training” process of a neural network with 2 “hidden” layers and several configurations

	N1	N2	MSE
	3	8	3.55×10^{-12}
	4	10	4.07×10^{-13}
	5	7	4.08×10^{-13}
	5	10	6.05×10^{-12}
	6	7	6.8×10^{-12}
	6	9	1.82×10^{-12}
	6	10	1.43×10^{-13}
	7	6	4.43×10^{-12}
	8	6	1.42×10^{-12}
	8	7	1.05×10^{-12}
The optimal configuration proved to be 6/10	9	7	7.55×10^{-12}
	10	5	8.61×10^{-12}
	10	10	2.74×10^{-12}

Table 2 Results of the “training” process of a neural network (50, 1, 2) with a single hidden layer and a configuration of 5 nodes in which the A_j data were computed

BASE: 45 curves, 50 input data
 %TRAINING/VALIDATION/TESTING: 80/10/10
 ALGORITHM: Levenverg-Marquard
 MSE TRAINING: 4.39595×10^{-8}
 MSE VALIDATION: 2.45985×10^{-8}
 MSE TESTING: 2.36959×10^{-8}
 EPOCHS: 100
 TIME/s: 0:00:04
 GRADIENT: 3.7×10^{-5}
 μ : 10^{-7}

“training”, “validation” and “testing” chosen were in all cases 80/10/10, and (e) the suitable optimization algorithm used was that of Levenverg-Marquardt (LM).

Table 2 shows the results of the “training” process of a neural network (50, 1, 2) with a single “hidden” layer and a configuration of 5 nodes in which the A_j data were computed.

The profiles of the output/target regression lines are shown in Fig. 4, corresponding to the processes of (a) “training”, (b) “validation” (c) “testing” and (d) jointly all processes simultaneously. The results obtained point to satisfactory MSE values (Table 2) and orders of magnitude similar to those of the *noise* errors of the data. The residuals distribution shows acceptable maximum fluctuations and the regression lines have equations with slopes close to unity, significantly null ordinates at the origin, and very good values (0.99998) of the Regression coefficients (R).

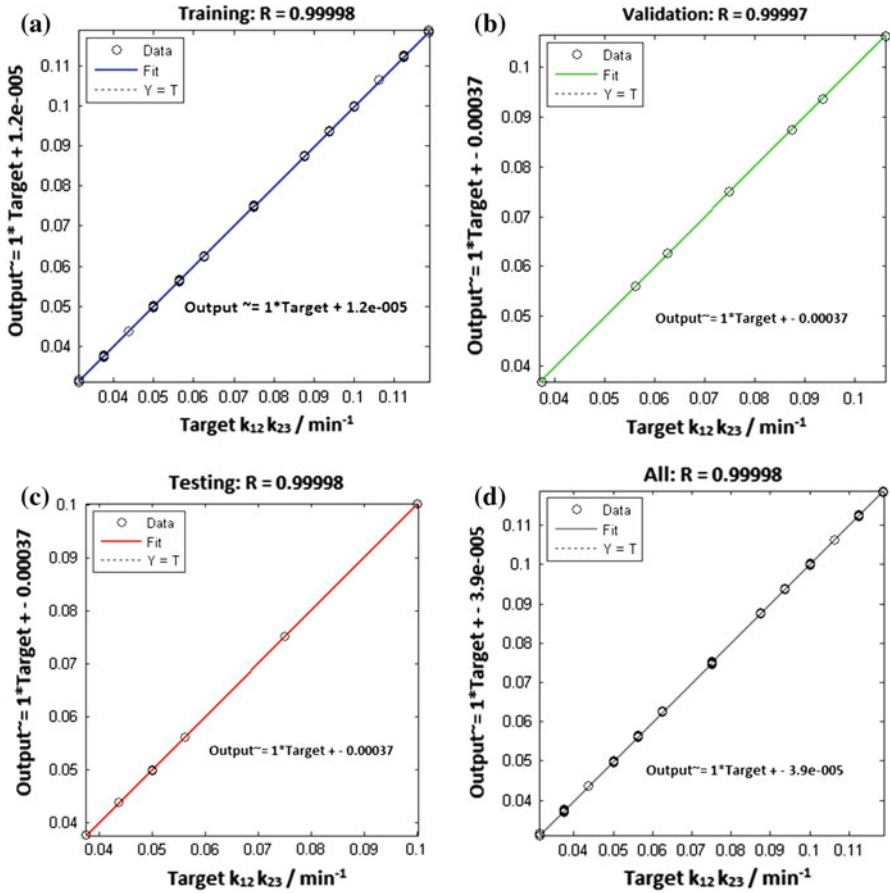


Fig 4 Plots of the representation of the Regression lines outputs/targets of the sets of curves, for the a “training” process b “validation” process c “testing” process and d jointly representation for the “training”, “validation”, and “testing” processes

4.3 Results of the prediction of the ANN

4.3.1 Synthetic kinetic data

In the study of the process of “prediction” of k_{12} and k_{23} from the synthetic data we quantitatively evaluated previously the influence of important variables such as the value of the *noise*, the number of curves to be predicted and the minimum number of species necessary for the “prediction”. This is possible with synthetic data, since the values of the parameters are known whereas in the case of experimental data the values of the parameters to be predicted are obviously unknown. The influences studied were as follows:

(a) The *noise* value: we imposed *a priori* different *noise* values on the synthetic data and determined the correlation and the tendency to variation of the error (SD) of

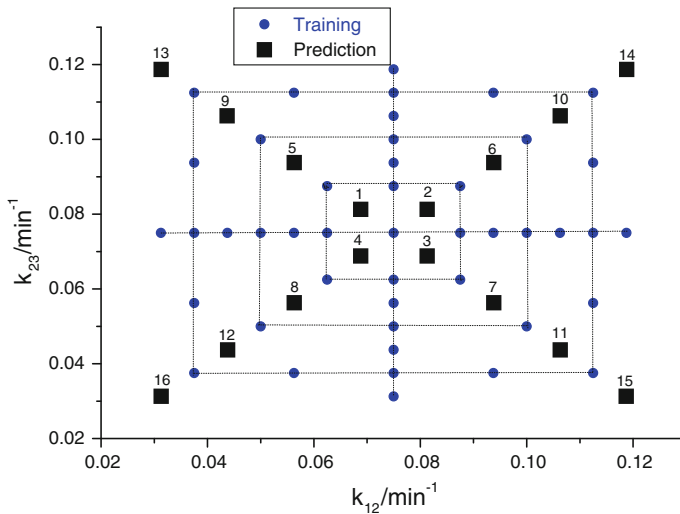


Fig 5 Graphic representation of the Experimental design (CSCED), where the 16 “prediction” points (1–16) are plotted jointly with the 45 points used for the process of “training” of the neural network

k_{12} and k_{23} for the different *noise* error values. It is not necessary to have a rigorous knowledge of the equation of the SD vs. *noise* regression lines; it suffices to know the individual SD values for the working *noise* and of the limits of the “experimental domain” of the *factors* in the maximum range of variation of the ED. These values were:

$$k_{12} = (3.1310^{-2} \pm 36 \cdot 10^{-4}) \text{ min}^{-1} \quad \text{and} \quad k_{23} = (1.188 \cdot 10^{-2} \pm 25 \cdot 10^{-4}) \text{ min}^{-1}$$

$$k_{12} = (1.188 \cdot 10^{-2} \pm 36 \cdot 10^{-4}) \text{ min}^{-1} \quad \text{and} \quad k_{23} = (1.188 \cdot 10^{-2} \pm 25 \cdot 10^{-4}) \text{ min}^{-1}$$

(b) Number of input curves: we determined the optimum number of curves for the “prediction” process, generating synthetic data of A_T , A_j , $[B_j]$ and α_j from points (k_{12}, k_{23}) of the ED different (not coincident) from those used in the “training” process corresponding to points 1–16 of Fig. 5. The plot of the SD values of both rate constants determined in the “prediction” processes on varying the number of curves chosen is shown in Fig. 6. It may be seen that in general upon increasing the number of curves moderately, the value of SD decreases for both rate constants down to a certain threshold value, after which the SD values show a tendency to decrease slightly or remain constant. This value is the optimal one and proved to be in the 8–12 curve range when starting from a base of 45 input curves.

(c) Minimum number of species monitored: we analyzed the possibility of the “prediction” of the rate constants when only 2 species or only a single one contributed to the kinetic data and compared the results with those obtained when the contribution was from the three species. This is a common situation, both when in the spectrum not all the species absorb and when the absorption of one of them predominates over those of the others and the relative values of the molar absorption coefficients are very different. On computing the kinetic data provided by only 2 species, in general

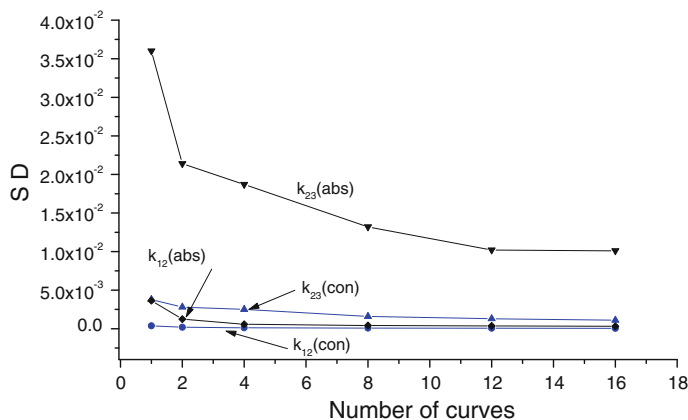


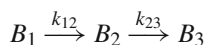
Fig 6 Plot of the SD values obtained in the “prediction” processes of both rate constants (k_{12} and k_{23}) versus the number of curves when both absorbance and concentration data are computed

results similar to those obtained by means of ANN computation of the data in which the 3 species contribute were obtained. This situation was also found in the case of the computation of kinetic data from a single chemical species if this corresponds to the intermediate species (B_2) since it is the one involved in both consecutive reactions.

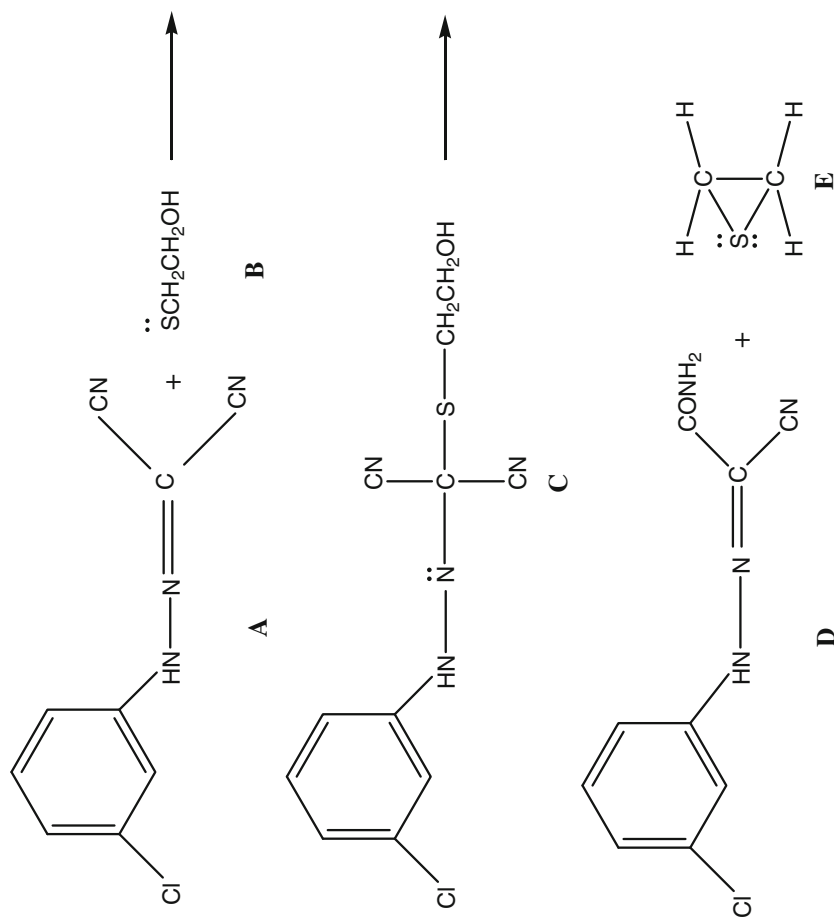
4.3.2 Experimental kinetic data

We performed the ANN treatment to the measured data from an experimental kinetic of a chemical reaction taking place in the laboratory. This was the reaction between *carbonyl cyanide 3-chlorophenylhydrazone* (3-Cl)-PHPD (**A**) with 2-mercaptoethanol (**B**). The kinetics of this reaction has been widely studied in the literature [25–27], where the various authors used different classic kinetics methods with concordant results. The reaction occurs according to a kinetic system that involves two irreversible consecutive reactions (Scheme 2). In the first step, an intermediate adduct (**C**) is formed, which is then hydrolyzed in an apparently intramolecular reaction to give the product 3-chlorophenylhydrazonocynoacetamide, (3-Cl)-PHCA (**D**) and the subproduct ethylene sulphide (**E**).

Since the reaction is carried out in an excess of the reagent (**B**), pseudo-first-order can be assumed and it is possible to express this schematically according to IUPAC norms [20]:



where species (**A**) is represented by B_1 , (**C**) by B_2 and (**D**) by B_3 . Chau *et al.* [26] performed a classic kinetic study based on data from the total absorbance (A_T) of the reacting mixture, monitored at 3 wavelengths (350, 375 and 400 nm), (Fig. 7) when the reaction conditions are, temperature of 20.0 °C and pH=4.3. We have used these kinetic data for the computational treatment with the approach combining ANNs and experimental design.



Scheme 2 Mechanism of the reaction between carbonyl cyanide 3-chlorophenylhydrazone (3-Cl)-PPHD with 2-mercaptoethanol

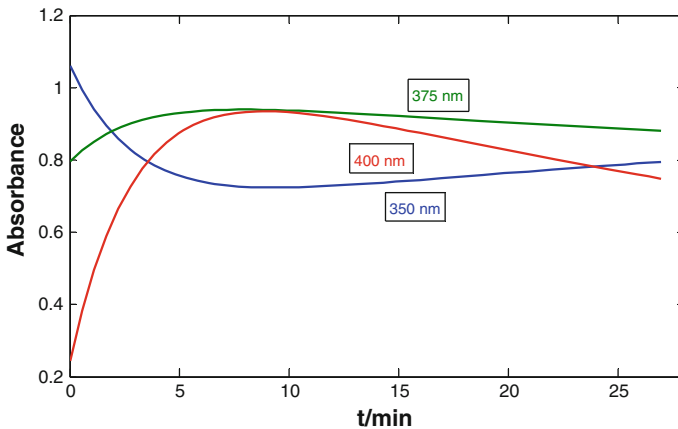


Fig 7 Profiles of the values of A_T with time acquired to 3 wavelengths (350, 375 and 400 nm) for the reaction between (3-Cl)-PPHD with 2-mercaptoethanol

The procedure of the ANN treatment carried out by us has the same steps as in the case of the synthetic data with *noise*; that is, the choice of a suitable ED, “training” of the neural network, evaluating the effect of the variables that provide the ideal conditions of the process and the optimal architecture of the neural network, and, finally, the “prediction” of the unknown values of both kinetic constants (k_{12} and k_{23}) whose values we aimed to determine. The characteristics that made the “training” process optimal were logically almost identical to those used in the case of the synthetic data. Thus, the most suitable “experimental design” was the same; that is, it was of the *CSCED* type. Bearing in mind the values of the initial concentration and those of the molar absorption coefficients of each species at each wavelength, we distributed the 45 curves of the base in 9 levels and 4 sub-levels and an “experimental domain” in the 0.35–0.02 range for both *factors*. As in the case of the synthetic data, each of the 45 curves forming the input base had 50 data of total absorbance (A_{T,t_i}^λ), with a significant range of fluctuation ($\approx 50\%$) and $\xi_{\max} = 0.85\xi$, in all cases using the 80/10/10 curve percentage ratio for “training”, validation” and “testing”. To determine the optimal architecture of the neural network, must be subjected to a new treatment. To determine the optimal structure and configuration of the hidden layer/s, we first considered an ANN with a single hidden layer (50, 1, 2), assaying a broad range of cases of configurations with different numbers of neurons. In all cases, the MSE values concerning “validation” and “testing” proved to be high (around $10^{-3} - 10^{-4}$) and hence unacceptable. Neither were acceptable the results of the regression lines of the output/target for each configuration nor the “performance” profiles of the curves of the three blocks. Accordingly, it was necessary to rule out the neural network formed by a single “hidden” layer and we tested a neural network formed by 2 “hidden” layers (50, 2, 2) and different configurations for both. The results obtained for the configurations (N1/N2) that provided the minimum validation values at the three working wavelengths (6/10) can be seen in Table 3.

Table 3 Results of neural network formed by 2 hidden layers (50, 2, 2) and different configurations when experimental data of the reaction between (3-Cl)-PHPD with 2-mercaptoethanol acquired at 3 wavelengths, have been computed

N1/N2	λ/nm		
	350	375	400
3/8	6.37E-7	2.40E-8	1.20E-7
5/7	2.58E-10	4.80E-9	1.85E-7
4/10	8.00E-8	1.72E-9	1.24E-7
5/10	2.63E-9	3.41E-9	2.25E-8
6/10	4.31E-9	1.42E-11	5.23E-9

In light of these results, it may be concluded that the optimal architecture of ANNs for the treatment of experimental kinetic data from the reaction under study has two different N1/N2 configurations, depending on the working wavelength. Thus, when the absorbance data were measured at $\lambda = 350\text{nm}$ the neural network was (50, 2, 2), 5/7 being the optimal configuration of the “hidden” layers, and when the data were acquired at 375 and 400 nm, the neural network was (50, 2, 2), with an optimal configuration of 6/10. Thus, we were able to predict the unknown values of the rate constants (k_{12} and k_{23}) by applying these two types of architecture and topology of the neural network depending on the value of λ . In all the “predictions”, we carried out a study of *Residual Analysis* and *Goodness of Fit*, determining the standard deviation (SD) of the fit with the expression:

$$SD = \left(\frac{\sum_{i=1}^{n_d} \sum_{j=1}^{n_c} (A_{ij}^{exp} - A_{ij}^{calc})^2}{n_d \cdot n_c} \right)^{1/2} \quad (9)$$

where A_{ij}^{exp} are the absorbance values of the input matrix and A_{ij}^{calc} are the absorbance values recalculated with the values of the kinetic constants obtained in the “prediction” process; n_d is the total number of data on the absorbance of the input matrix, and n_c is the number of curves.

Table 4 shows the results obtained in the “prediction” process; that is, the values of the k_{12} and k_{23} rate constants together with the SD errors obtained from the Residual analysis after combined treatment of ANNs and Experimental Design of the data on the reaction under study acquired at each of the 3 wavelengths. It may be seen that the results obtained with the “prediction” for the values of the rate constants obtained to the 3 wavelengths point to that coincide satisfactorily. This is proof in favor of the suitability and applicability of the ANN method for the kinetic treatment and consequent determination of the kinetic parameters. The mean values calculated from the individual values at the three wavelengths, for both kinetic constants at 20.0 °C and pH = 4.3, are those corresponding to the kinetics of the reaction studied. These mean values ($\langle k_{12} \rangle = 0.3381 \text{ min}^{-1}$ and $\langle k_{23} \rangle = 0.0195 \text{ min}^{-1}$) coincide with

Table 4 Values of the k_{12} and k_{23} rate constants together with the SD errors obtained after the combined treatment of ANNs and experimental design in the “prediction” process of the experimental data on the reaction between (3-Cl)-PHPD with 2-mercaptoethanol acquired at 3 wavelengths

λ/nm	k_{12}/min^{-1}	k_{23}/min^{-1}	SD
350	0.3332	0.020 ₄	9.26E–4
375	0.3359	0.020 ₂	7.70E–3
400	0.3454	0.017 ₉	6.10E–3
450	0.3672	0.031 ₈	5.92E–4

those obtained by other authors [25–27] who have independently studied the kinetics of this chemical reaction by means of classical kinetic methodologies.

References

- G. Puxty, Y.M. Neuhold, M. Jecklin, M. Ehly, P. Gemperline, A. Nordon, D. Littlejohn, K. Basford, M. De Cecco, K. Hungerbühler, *Chem. Eng. Sci.* **63**, 4800–4809 (2008)
- A. De Juan, E. Casassas, R. Tauler, *Encyclopedia of Analytical Chemistry: Instrumentation and Applications. ‘Soft-modelling of analytical data’* (Wiley, New York, 2000)
- A. De Juan, M. Maeder, M. Martínez, R. Tauler, *Chemom. Intell. Lab. Syst.* **54**, 123–141 (2000)
- J. Casado, J.L. González, M.N. Moreno, *React. Kinet. Catal. Lett.* **33**, 357–362 (1987)
- J. Casado, J.L. González, M.N. Moreno, G. Sánchez, *React. Kinet. Catal. Lett.* **36**, 337–344 (1988)
- J. Havel, J.L. González, P. Palacio, *Script. Chem.* **22**, 27–33 (1992)
- J. Havel, F. Cuesta, J.L. González, M.M. Canedo, *J. Anal. Chem.* **51**, 110–115 (1996)
- J.L. González-Hernández, M.M. Canedo, A. Domínguez-Gil, J.M. Lanao, *J. Pharm. Sci.* **81**, 592–596 (1992)
- M.M. Canedo, J.L. González-Hernández, *Chemom. Intell. Lab. Syst.* **66**, 63–78 (2003)
- J.L. González, M.M. Canedo, C. Grande, *Int. J. Chem. Kinet.* **38**, 38–47 (2006)
- M.M. Canedo, J.L. González-Hernández, *J. Math. Chem.* **49**, 163–184 (2011)
- MathWorks MatLab R2012a, Vs7.14.0.739 (2012)
- F. Amato, J.L. González-Hernández, J. Havel, *Talanta* **93**, 72–78 (2012)
- S. Ventura, M. Silva, D. Pérez-Bendito, C. Hervás, *Anal. Chem.* **67**, 1521–1525 (1995)
- C. Hervás, S. Ventura, M. Silva, D. Pérez-Bendito, *J. Chem. Inf. Comput. Sci.* **38**, 1119–1124 (1998)
- S.J. Ardakani, F. Gobal, *React. Kinet. Catal. Lett.* **85**, 347–382 (2005)
- P. Valeh-e-Sheyda, F. Yaripour, G. Moradi, M. Saber, *Ind. Eng. Chem. Res.* **49**, 4620–4626 (2010)
- B. Kovacs, J. Tóth, *Int. J. Appl. Math. Comput. Sci.* **4**, 7–11 (2007)
- N.H.T. Lemes, E. Borges, J.P. Braga, *Chemom. Intell. Lab. Syst.* **96**, 84–87 (2009)
- K.J. Laidler, *Pure Appl. Chem.* **68**, 149–192 (1996)
- M.N. Berberan-Santos, J.M.G. Martinho, *J. Chem. Ed.* **67**, 375–379 (1990)
- Kohonen T, *An Introduction to Neural Computing*, ed. Neural Networks (1988)
- S. Curteanu, H. Cartwright, *J. Chemom.* **25**, 527–549 (2011)
- G.M. Vandeginste, D.L. Massart, L.M.C. Buydens, S. de Jong, P.J. Lewi, J. Smeyers-Verbeke, *Handbook of Chemometrics and Qualimetrics* (Ed. Elsevier, Amsterdam, 1998)
- R.H. Bisby, E.W.K. Thomas, *J. Chem. Ed.* **63**, 990–992 (1986)
- F.T. Chau, K.W. Mok, *Comput. Chem.* **16**, 239–242 (1992)
- S. Bijlsma, Louwerse (Ad) D.J., Windig W., Smilde A.K. J. *Chemom.* **13**, 311–319 (1999)

A New Computational Treatment in Non-Isothermal Chemical Kinetics by Application of a Robust Hybrid Algorithm

S. Encinar, J. L. González-Hernández*, M. Mar Canedo

*Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca,
E-37008 Salamanca, Spain*

(Received January 13, 2014)

Abstract.

In the present work we designed and successfully applied a new Hybrid Algorithm (HA) able to optimize Activation Thermodynamic Parameters (ATP) directly from the computational treatment of non-isothermal kinetic data. It comprises two methods based in different mathematical principles whose sequential application is carried out. Firstly it uses a “*soft modeling*” method of Artificial Neural Networks (ANN) that does not need to use the *initial estimates* of the parameters and determine the ATPs values in the neighborhood of the minimum global. These values are improved later after application of a mathematical optimization method based in a second order gradient algorithm (AGDC) able to reach the desired global optimum.

The application of HA that we propose provided good and satisfactory results for the optimized values of the ATPs. HA offers several advantages with respect to the isothermal classic methods since it allows to optimize ATPs directly without the need to determine the kinetic constants previously. In addition for the case of non-isothermal experiments, a single replicated kinetic experiment is enough since it allows the computation of a larger number of kinetic data that implies a considerable saving of reagents and laboratory time.

(*) *Author for the correspondence*
e-mail: jlg93@usal.es

1. Introduction

Determination of the parameters involved in the mathematical functions by means of the application of computational treatments is usually carried out by means of mathematical optimization methods applying gradient algorithms. These non-linear iterative fitting techniques are often used in the various fields of Chemistry, in particular, and of Science in general. However, a major drawback of these optimization methods is their high sensitivity to the *initial estimates* of the parameters to be supplied. Only if these values are very close to the global minimum can a rapid and reliable convergence of the iterative process be expected, guaranteeing success of the parameter optimization. If the *initial estimates* are far from the global minimum the process may become divergent or may reach a singular point (*local minimum, saddle point, etc...*), leading the optimization process to fail. This often happens in the treatment of kinetic models. Additionally there is a series of problems involving *identifiability* and *distinguishability*, which lead to different types of ambiguity in the solutions to the *stiff* systems of ordinary differential equations (ODE). In light of this, it would seem appropriate to design and apply a new method that, initially, would provide an approach to the global optimum and then use such results as a starting point to apply a robust gradient method that will guarantee the success of the mathematical optimization of parameters.

In the present work we designed and successfully applied a new Hybrid Algorithm (HA) able to optimize Activation Thermodynamic Parameters (ATP) -Pre-exponential factor (A) and the Activation Energy (Ea)- directly from non-isothermal kinetic data. It comprises two methods based in different mathematical principles whose sequential application is carried out. Firstly we applied a "*soft modeling*" method using Artificial Neural Networks (ANN) that does not need to use the *initial estimates* of the parameters and its goal is the determination of the parameter values (*outputs*) in the neighborhood of the optimum global. These values will be the *initial estimates* of a robust and efficient second order gradient algorithm (AGDC) [1] able to reach the desired global optimum that will guarantee the success of the final optimization of the values of the parameters. The classic method used to determine ATPs consists of applying "*hard-modelling*" techniques to series of data of isothermal kinetic experiments acquired at different temperatures to initially obtain the rate constants of each kinetic experiment. With the values of these, the ATPs are obtained in a second step from the linearized Arrhenius equation.

The application of this HA that we propose, provided several advantages since it became possible to optimize those parameters directly, without the need to determine the

kinetic constants in a previous step. To be able to apply this treatment, it was necessary to acquire the kinetic data from non-isothermal kinetic experiments [2], imposing a controlled variation of temperature along the reaction kinetic. The number of the set of kinetic data computed in the classic isothermal procedure ($T=\text{constant}$) is very small since it is limited by the experimental requirements when dealing with a low number of isothermal experiments (normally, 10-12), each of them performed at a different temperature. In the case of non-isothermal experiments, a single replicate kinetic experiment is enough since it allows the computation of a huge set of kinetic data which, bearing in mind the laboratory time and reagents saved, is a great advantage.

The literature contains several recent references in which the authors have applied Hybrid Algorithms (HA) in the field of Chemistry and that essentially comprise Genetic Algorithms (GA). Their design and application are currently able to solve problems pending solution and reveal the high degree of reliability and precision in the results obtained. Of interest is one review [3] in which the authors consider a broad range of applications of different types of GA in Chemometrics: M. Maeder *et col.* [4] determine the rate and equilibrium constants of reaction mechanisms by application of a Hybrid Algorithm based on a Genetic Algorithm. C. Hervás *et col.* [5] use GA and pruning computational neural networks for selecting the number of inputs required to correct temperature variations in kinetic-based determinations. Artificial Neural Networks (ANN) offer a versatile “*soft modeling*” method that can be applied in diverse fields with acceptable results [6]. The method is applied for quantitative purposes, among others, in the so-called *Principal Component Analysis (PCA)* method, in which there are no explicit functions of multivariate correlation or, if there are, they are extraordinarily complex. The results of applying ANN for quantitative purposes in chemical kinetics in simple models [7,8] are in some cases acceptable. However, it is necessary to perform an exhaustive process of *training* of the neural network in order to obtain its optimal architectures, meaning that the method is time-consuming and tedious and that it cannot be used individually in all cases. Nevertheless, it is an ideal method for carrying out the approximation of the ATPs values to those of the global minimum, without previously knowing anything about the magnitude and sign of the parameters. The AGDC algorithm was designed and successfully implemented at our laboratory in the treatment of many fields within the area of Physical Chemistry [9-13]. It is a second-order gradient algorithm in which continuous control of the direction and sense of the movement vector is carried out, evaluating each step performed in the iterative process. It is endowed with corrective devices that, when necessary, are able to correct and modify the movement vector automatically in

order to ensure the decrease to reach the minimum global. Additionally, it is refractive to the presence of local minima and *ill-conditioned* response surfaces and includes different graphic representation methods (3D, contour maps, etc.) that allow both the graphic follow-up of the global minimum and a detailed exploration of the region of the global minimum to detect the existence of singular points.

2. Theoretical and Computational Aspects

2.1. Chemical Kinetic Aspects

Let us consider in general a chemical system formed by n_r chemical elementary (or concerted) reactions where n_s chemical species can be involved. According to IUPAC's norms [14] the r -th chemical reaction can be expressed for the generic equation

$$0 = \sum_{j=1}^{n_s} \nu_{j,r} B_j \quad (1)$$

where, B_j is the chemical species involved in the system of reactions; $r=(1, \dots, n_r)$, the number of chemical reactions; $j=(1, \dots, n_s)$, the number of chemical species; $\nu_{j,r}$, the stoichiometric coefficient of the species B_j in the r -th reaction; $\nu_{j,r} < 0$ when B_j plays only the role of reactant in the r -th reaction and $\nu_{j,r} > 0$ when B_j plays only the role of product in the r -th reaction.

When the reaction is an elementary or concerted one, the absolute values of the kinetic order ($z_{i,r}$) and stoichiometric coefficient of B_j coincide, that is $|\nu_{i,r}| = |z_{i,r}|$. The rate differential equation of the chemical species B_j in the r -th is given by

$$d [B_j] / dt = k_r \nu_{j,r} \prod_{i=1}^{n_s} [B_i]^{|z_{i,r}|} \quad (2)$$

where B_i are the species playing only the role of reactants in the r -th reaction ($\nu_{i,r} < 0$) and k_r is the kinetic rate constant of the r -th reaction. Each chemical species can take part in several reactions and the rate differential equations will be the sum extended over those reactions where the reactant B_i appears, obtaining a system of ordinary differential equations (ODE) according to the generic equation,

$$d [B_j] / dt = \sum_{r=1}^{n_r} k_r \nu_{j,r} \prod_{i=1}^{n_s} [B_i]^{|z_{i,r}|} \quad (3)$$

The general solution of the system of rate ODE give the explicit function of the concentrations of the all species with the time ($[B_j]_t$). If the experimental data are expressed in absorbance, we have to consider the Lambert-Beer-Bouguer law:

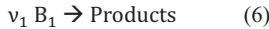
$$A_{j,t_i}^\lambda = \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad (4)$$

where A_{j,t_i}^λ is the absorbance of the species B_j at the, time t_i and path length 1 cm and ε_j^λ is the molar absorption coefficient of B_j at the wavelength λ . The absorbance of the mixture (A_{T,t_i}^λ) measured at wavelength λ , time t_i and temperature T and, can be expressed as:

$$A_{T,t_i}^\lambda = \sum_{j=1}^{n_s} A_{j,t_i}^\lambda = \sum_{j=1}^{n_s} \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad (5)$$

2.2. Thermodynamic Aspects in Non-Isothermal Kinetics

Let us consider the simple chemical reaction:



The rate differential equation (3) corresponding to this reaction, expressed in function of the *extent of reaction* variable in units of molar concentration (ξ') when the reactant B_1 is of kinetic first order, can be written as:

$$d\xi'/dt = k(T)([B_1]_0 - |v_1|\xi') \quad (7)$$

Separating variables and later integration of the first member of the equation, we have

$$-(1/|v_1|)\ln\{([B_1]_0 - |v_1|\xi')/[B_1]_0\} = \int_0^t k(T)dt \quad (8)$$

The Activation Energy, (E_a) and the Pre-exponential Factor (A) can be determined substituting $k(T)$ in (8) according the Arrhenius equation, where α is the remaining molar fraction of the reactant B_1 .

$$-(1/|v_1|)\ln \alpha = \int_0^t A e^{-E_a/RT} dt \quad (9)$$

In non-isothermal conditions, the second member of (9) cannot be directly integrated since there are two dependent variables (T and t). It will be crucial to establish the identity of this function because the mathematical method of resolution of the equation (9) will be different. The function must be monotonically increasing, since is this way it will be possible to minimize the great differences in the reaction rate existing between the beginning and the end of the reaction; the rate of heating must be suitable for the interval of time studied and its profile must be reproducible in the laboratory.

We consider several possibilities for the function $T = f(t)$

1.- Inverse function of the temperature corresponding to a hyperbolic branch

$$1/T = (1/T_0) - mt \quad (10)$$

Substituted in (9), we have

$$-(1/|v_1|)\ln \alpha = \int_0^t A e^{-E_a/R[(1/T_0)-mt]} dt \quad (11)$$

This function has a great advantage since the integral of (9) has a known primitive function and therefore, mathematical exact solution with the following expression:

$$\alpha = e^{-[(|v_1|A R/m Ea)(e^{-Ea/RT_0})(e^{(m Ea/R) t} - 1)]} \quad (12)$$

We have to take in account the following important considerations:

a) the explicit function of α depending on the time (12) is not simple and there is necessary the application of a method of sufficiently robust treatment for the determination of the ATPs. In addition it is necessary to consider the great difference in the order of magnitude of both parameters that complicates extraordinarily the success in the application of the method of treatment.

b) the experimental points (T/t) must belong to this inverse hyperbolic function and therefore, it is necessary to reproduce the profile of the curve of the function (T/t) (10) with the points obtained in the laboratory.

2.- Function T = f(t) of polynomial type of n-th degree

$$T = \sum_{j=0}^{j=n} a_j t^j \quad (13)$$

substituting in (8) according the Arrhenius equation,

$$-(1/|v_1|) \ln \alpha = \int_0^t A e^{-Ea/R [(1/\sum_{j=0}^{j=n} a_j t^j)]} dt \quad (14)$$

This equation does not have mathematical exact solution since the integration cannot be performed. We have 2 options of treatment:

2.1- Numerical integration of the equation (14) using appropriate “*quadrature formulas*” corresponding to a suitable numerical algorithms of resolution of integrals (Simpson, Lobatto, Gauss-Kronrod, Vectorized, etc.)

2.2- Numerical resolution directly from the beginning of the ordinary differential equation (eq. [7]) expressed in terms of Arrhenius’s equation, that is:

$$d[B_i] = -[B_i]A e^{-Ea/RT} dt \quad (15)$$

The solution of this differential equation (15) can be performed by means of the application of numerical suitable methods that they must be suitable for the treatment of “*stiff*” systems (Runge-Kutta methods, Gear's method (BDFs), Rosenbrock formula, trapezoidal rule with "free" interpolant....etc).

2.3. Mathematical Aspects. Hybrid Algorithm (HA)

The Hybrid Algorithm (HA) designed in our laboratory comprises two methods based in different mathematical principles and it is applied sequentially in two steps. In the first, a method of

Artificial Neural Network (ANN) is applied. Its most important feature, and indeed a great advantage, is that it is not necessary to supply *initial estimates* values for the determination of ATPs (A and Ea). The results obtained in the output matrix are near to those that belong to the global optimum sought. These values are precisely the *initial estimates* for starting the process of ATPs optimization via application of the AGDC algorithm, constituting the second step in the treatment. This is an improvement of the values obtained in the output matrix after the application of the ANN, thus reaching the global minimum that will guarantee the determination of the optimized values of the ATPs.

2.3.1. Artificial Neural Networks (ANN)

Artificial Neural Networks are parallel interconnected networks of simple computational elements called neurons and are structured in layers that are intended to interact with the objects of the real world in a similar way to the biological nervous systems [15]. Parallel processing is the ability of the brain to simultaneously process incoming stimuli of differing quality. The multilayer neural network uses sets of input data and parameters (called *targets*), distributed in 2 input matrices when Matlab [16] is applied. The elements of the input matrix are the calculated synthetic values, where one row contains a single curve of the data and all the curves thus obtained (n_c) are grouped in an *input data* matrix. The *target matrix* is formed by the sets of parameters (n_p). In our case, the input data matrix contained the kinetic data of all curves ($\alpha_j, A_T, A_j, [B_{ij}]$) and the *target matrix* ($n_c \times n_p$) contained the set of kinetic rate constants (k_{mm}). Formally, a multilayer neural network is an oriented graph in which the nodes represent a set of processing units, called neurons, and the connections represent the information flow channels. Each connection between two neurons has an associated value called “*weight*” which specifies the strength of the connection between neurons. Positive and negative values determine excitatory and inhibitory connections, respectively. The choice of a specific class of networks for the approximation of a nonlinear map depends on a variety of factors dictated by the context and is related to the desired accuracy and the prior information available concerning the input-output pairs.

The first layer of a multilayer neural network contains neurons that receive the input data values from the elements of the input data matrices. This information is transmitted from the i -th neuron of a layer to the j -th neuron of the subsequent one, with a weight w_{ji} . A neuron parameter (*bias*) is summed with the weighted inputs of the neurons and passed through the transfer function to generate the output of the neurons.

The layer following the input one is called *hidden*. In each neuron of a *hidden* layer the weighed inputs coming from the previous one are summed with each other and added to a *bias*. The result is then transformed by means of a suitable mathematical function to obtain an output called *activation of the neuron*, which is transferred to the neurons in the next layer after another weighing step. The output parameters values are calculated in the last layer (*output layer*) by means of a suitable transformation function.

The process described is called to as the *training* or *learning* of the multilayer neural network and constitutes an iterative method where the iterations are called *epochs*. After each *epoch*, the calculated values of the parameters are grouped in the *output matrix* (b_{ij}^{output}) and they are compared with those of the corresponding curve in the *target matrix* (b_{ij}^{target}) and the optimum value of the Mean Squared Error (MSE), expressed in absolute value, is calculated according the following equation:

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{output} - b_{ij}^{target})^2}{n_p \cdot n_c} \right)^{1/2} \quad (16)$$

where n_c is the input number of curves and n_p is the number of parameters, $n_c \cdot n_p$ being the dimensions of both matrices (*output matrix* and *target matrix*).

During the process of *training*, *weights* and *bias* values are modified with suitable mathematical optimization algorithms in order to minimize the calculated values of MSE in each *epoch*. In the present work, the *back-propagation* algorithm was used. The iterative process finishes when the minimum value of MSE is reached, after which the *training* process can be considered to be completed.

It is necessary to know the optimal architecture and topology of the multilayer neural network in order to obtain the best results when ANN is applied to the system under study. We have used a method of *trial and error* by minimizing the MSE values obtained for the different possible configurations of the same number of *hidden* layer/s. It must to determine the minimum value (optimum) of the MSE for all possible configurations for the *hidden* layer/s chosen. For each *hidden* layer, a graph of MSE values vs. the number of neurons shows that initially, for the lower configurations, the value of the MSE decreases rapidly when the number of neurons increases, but after a constant value or a poor improvement is obtained. The optimum number of neurons (*configuration*) in that *hidden* layer is given by the point of intersection of the two branches of the graph. Sometimes, a small minimum appears near this intersection point. The architecture of the neural network can be written in

abbreviated notation as $(n_{inp}, n_{hid}, n_{out})$, where n_{inp} is the number of neurons in the *input* layer, n_{hid} in the *hidden* layer and n_{out} in the *output* layer.

Neural network *training* is completed with the processes of *validation* and *testing*. These are 2 control and verification processes of the iterative minimization method between the elements of the *output* and *target* matrices. Among the different curves comprising the *input* matrix, random choice is made of a percentage of the total, established previously (5%,10%...), which gives rise to a “*sub-matrix*” of input curves that are subjected to iterative optimization until a minimum MSE value is reached. It is thus possible to verify the validity of the *training* process by ensuring that it is convergent, that it has an appropriate termination, and that there not been any “*overfitting*”, since any possible “*overtraining*” has taken been into account. Validation is completed when in a given number (≥ 6) of consecutive *epochs* the MSE remains constant or shows a slight tendency to increase. The *testing* process is similar, except that the control for terminating the process is performed by controlling the computation time instead of the number of *epochs*.

The process of *prediction* consists of the determination of the unknown parameters from a set of experimental data after application of the optimal and trained neural network. Obviously, the elements of the *target* matrix are unknown for this *prediction* process, and only the *input data* matrix is provided to the neural network. In our case, the elements of the *input data* matrix in the process of *prediction* are experimental kinetic values ($\alpha_j, A_T, A_j, [B_j]$, etc.), acquired from a real system of reactions developed at the laboratory.

The computational work of ANN has been performed using Matlab environment [16] by means of the application of “*Neural Networks Toolbox*” with the creation of user’s interfaces (GUI) including the appropriate individual analysis of Residuals and errors (MSE, SD, etc) and special plotting programs for visualizations 3D of SQD functions.

2.3.2. AGDC Algorithm

Mathematical optimization is a computational method that allows the determination of different parameters by means of the maximization or minimization of the objective function. Usually performed objective function minimization using different methods: Search algorithms (Simplex, Fibonacci,..etc) or Gradient algorithms (Steepest Descent, Gauss-Newton, Levenberg-Marquardt, AGDC,...etc) [17]. In this paper we have used the AGDC mathematical optimization algorithm [13,18,19], this is a second-order gradient method that minimizes the numerical function Sum of Quadratic Deviations (SQD) and whose *initial estimates* are the *outputs* values obtained by ANN.

The procedure followed by AGDC, step by step, is the following:

1. Select the parameters to be optimized \mathbf{X} (Ea/A)
2. $m=0$ (m = iteration number). Input data:
 - 2.1. Experimental data of $(\alpha_i)_{exp}/t$, initial concentrations, Convergence Criterion (CC), etc.
 - 2.2. *Initial estimates* of the unknown parameters $\mathbf{X}^{(0)}$ (Values of *outputs* from ANN).
3. Determine the $SQD^{(0)}$ function.
 - 3.1. Calculate of concentrations $[B_1]_i$ by **a)** Mathematical exact solution [eq. (12)]; **b)** Numerical integration [eq.(14)] **c)** ODE [eq. (15)].
 - 3.2. Calculate $(\alpha_i)_{calc}$ $(\alpha_i)_{calc} = [B_1]_i/[B_1]_0$
 - 3.3. Calculate $SQD^{(0)}$ by the equation: $SQD(\mathbf{X}) = \sum_{i=1}^{N_d} \left((\alpha_i)_{calc} - (\alpha_i)_{exp} \right)^2$

4. AGDC ALGORITHM

- 4.1. Calculate the vector of movement $\mathbf{p}^{(m)}$.
 - 4.1.1. Compute partial numerical derivatives of $(\alpha_i)_{calc}$ with respect to the parameters to be determined $\mathbf{X}^{(m)}$, $(\partial(\alpha_i)/\partial X_p)$
 - 4.1.2. Compute Gradient vector and Hessian Matrix ($\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$).
 - 4.1.3. Compute $(\mathbf{H}^{(m)})^{-1}$
 - 4.1.4. Calculate the components of the vector of movement

$$\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}[\mathbf{H}^{(m)}]^{-1}$$
- 4.2. Control and correction of the direction of the vector of movement $\mathbf{p}^{(m)}$
 - 4.2.1. If $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$, go to 4.3.
 - 4.2.2. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} < \varepsilon$ (ε = scalar close to zero), $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ and go to 4.3
 - 4.2.3. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} > 0$, $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$
- 4.3. Control the length of the vector of movement $\mathbf{p}^{(m)}$.
 - 4.3.1. Compute the scalar $(\lambda^{(m)})$ by the method of Hartley [17]
 - 4.3.2. $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \lambda^{(m)} \mathbf{p}^{(m)}$
 - 4.3.3. Determine the $SQD^{(m+1)}$ function.
 - 4.3.4. If the Goldstein-Armijo criterium [20] is satisfied go to 4.4.
 - 4.3.5. $\lambda^{(m)} = \lambda^{(m)}/2$ go to 4.3.2.

4.4. Calculate

$$CON = \left| \frac{SQD^{(m+1)} - SQD^{(m)}}{SQD^{(m)}} \right|$$

- 4.5. If convergence is not attained ($CON > CC$), set $m = m + 1$ and go to 4.1.
5. Final of Optimization. $\mathbf{X}^{(m+1)} =$ Optimized Parameters.

5.1 Calculation of the errors of the parameters.

5.2 Statistical analysis of residuals: arithmetic mean, variance, standard deviation, square error, statistical measure of adjustment and Pearson function (χ^2) [21].

6. End of AGDC

7.- End of HA.

The elements constituting the Hessian matrices and gradient vectors are detailed in the following expressions:

$$\mathbf{g} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} RES_i \frac{\partial \alpha_i}{\partial E_a} \\ \sum_{i=1}^{N_d} RES_i \frac{\partial \alpha_i}{\partial A} \end{bmatrix} \quad \text{where,} \quad RES_i = [(\alpha_i)_{calc} - (\alpha_i)_{exp}]$$

$$\mathbf{H} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \left(\frac{\partial \alpha_i}{\partial E_a} \right)^2 & \sum_{i=1}^{N_d} \left(\frac{\partial \alpha_i}{\partial E_a} \right) \left(\frac{\partial \alpha_i}{\partial A} \right) \\ \sum_{i=1}^{N_d} \left(\frac{\partial \alpha_i}{\partial A} \right) \left(\frac{\partial \alpha_i}{\partial E_a} \right) & \sum_{i=1}^{N_d} \left(\frac{\partial \alpha_i}{\partial A} \right)^2 \end{bmatrix}$$

The computational program has been designed and performed in our laboratory using a computational executable code programs (##.m type), in the Matlab environment [16], using “m” language and it is constituted by a Main program and several Functions and/or Subroutines.

3. Kinetic Data. Input Curve Base

3.1. Study of the Functions T/t and α/t

The function T/t has to fulfill the following requirements: *a)* the function must be monotonically increasing, since is this way it will be possible to minimize the great differences in the reaction rate existing between the beginning and the end of the reaction when isothermal conditions are considered, *b)* the rate of heating must be suitable for reaching a convenient extent of reaction in the interval of time in which the non-isothermal

kinetic data will be acquired, and *c*) the function must be accessible to experimentation and accurately reproducible in the laboratory. The type of functions that we have chosen are:

1.- inverse function of the temperature corresponding to a hyperbolic branch [eq. (10)] that permits to reach an exact mathematical solution of the rate differential equation;

2.- function of polynomial type of *n*-th degree according equation (13), without of exact mathematical solution for the rate differential equation.

It is seen that as the value of “*m*” of equation (10) decreases a smoothing of the concavity of the hyperbolic branch is observed, tending towards linearity. The suitable value, and the one that best satisfies the 3 requisites considered above, is $4.2 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$. We fitted the T/t points to a first-degree polynomial function and then progressively to polynomials from 2nd to 5th degree, observing that the coefficients of the independent variable had negligible values with respect to the value of *a*₁ from the quadratic term. The coefficients (*a*₀, *a*₁, ...*a*₅) of the polynomials of 1st, 3rd and 5th degree obtained in the fittings are shown in Table 1.

Polynomial grade	<i>a</i> ₀	<i>a</i> ₁	<i>a</i> ₂	<i>a</i> ₃	<i>a</i> ₄	<i>a</i> ₅
1	298.15	0.3734	-	-	-	-
3	298.15	0.3734	5.0E-4	6.0E-7	-	-
5	298.15	0.3734	5.0E-4	6.0E-7	7.0E-10	1.0E-12

Table 1. Values of the coefficients of the polynomials functions of 1st, 3rd and 5th degree obtained in the fittings according equation (13).

The values of the Statistical Analysis of Residuals show that the linear fitting is correct. This means that the T/t data pairs generated with the inverse hyperbolic function with a value of $m=4.20 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$ can be satisfactorily fitted to a linear function. At the laboratory, this allowed us to impose a linear heating rate (slope of $0.3734 \text{ K min}^{-1}$), starting (t=0) at T= 298.15 K. Thus, it was guaranteed certain that the T/t experimental points would coincide with the exact solution of the differential rate equation (12). That is, it ensures satisfactory agreement between the kinetic values of the linear functions and those of the inverse hyperbolic branch.

The profiles of the non-isothermal kinetic curves (*α*/t) sometimes show concave down segments while the isothermal curves are always concave up. This can be explained in terms of the notion that in non-isothermal curves there are two opposing phenomena that affect the reaction rate: the increase in temperature with time increases the reaction rate, and the logical

decrease in the concentration of reagents as they are consumed reduces it. Depending on which of the two phenomena predominates, one concavity or the other will be observed and even when both effects are balanced quasi-linear profiles appear. This can be seen in Figure 1, where the curve (I) is a sigmoid corresponding to non-isothermal kinetics in which three behaviours are clearly visible. Segment (a) is concave down since the effect of temperature predominates and segment (c) is concave up since the effect of the temperature decrease and the concentration influence predominates. The intermediate segment (b) is linear because both effects are balanced. Curve (II) has a linear profile since both effects are equilibrated almost throughout the kinetics and the whole of curve (III) is concave up, as corresponds to an isothermal kinetics in which exclusively of the effect of concentration exists.

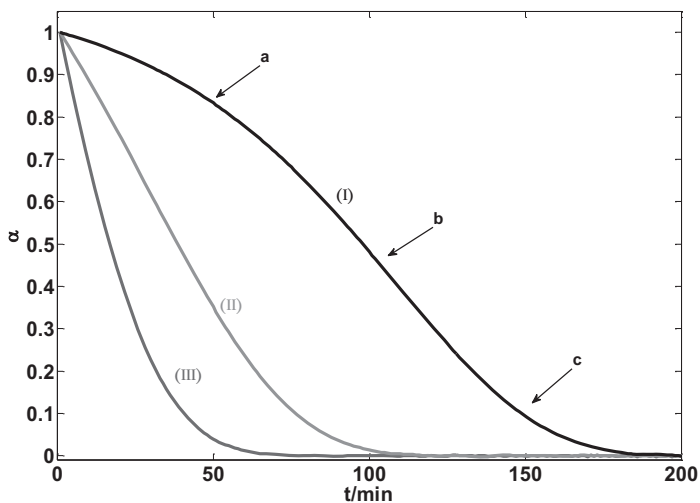


Figure 1. Several profiles of non-isothermal and isothermal kinetic curves (α/t). Curve (I) is a sigmoid corresponding to non-isothermal kinetic in which three behaviours are clearly visible: segment (a) is concave down since the effect of temperature predominates; segment (c) is concave up since the effect of the temperature decrease and the concentration influence predominates; segment (b) is linear because both effects are balanced. Curve (II) has a linear profile since both effects are equilibrated. Curve (III) is concave up, as corresponds to an isothermal kinetic in which exclusively of the effect of concentration exists.

3.2. Experimental Design (ED)

The design and implementation of a suitable Experimental Design (ED) is crucial to ensure the success of the *training* process of neural networks. We consider 2 *factors* (A and Ea) whose *responses* are the non-isothermal kinetic data of the base of the input curves (α/t). It is necessary to consider 2 variables: a) the extreme values of both *factors* that configure the *experimental domain* and b) their relative values. Both variables must ensure that the binary combinations of both *factors* will generate a set of kinetic curves that will have sufficient information to ensure an optimal *training* process of the neural network. In addition, the number of *levels* of the *factors* of the ED must be suitable if one is to avoid useless computational work and avoid large differences in the spacing of the values of the *responses*. Accordingly, to optimize the *training* process the kinetic curves of the *input matrix* must have efficient kinetic information and must be correctly distributed according to the choice of a suitable experimental design and an appropriate *experimental domain*. In agreement to the results of the study of the functions T/t, we have generated the non-isothermal kinetic data (α/t) in order to obtain the curves of the *input matrix* to perform the *training* process of the neural network from the ATPs organized according to a suitable ED distribution. Bearing in mind the characteristics of the ANN computation of the non-isothermal kinetic system under study, we have chosen as the optimum one, the ED corresponding to the *Central Star Composite Experimental Design (CSCED)* distribution represented in Figure 2.

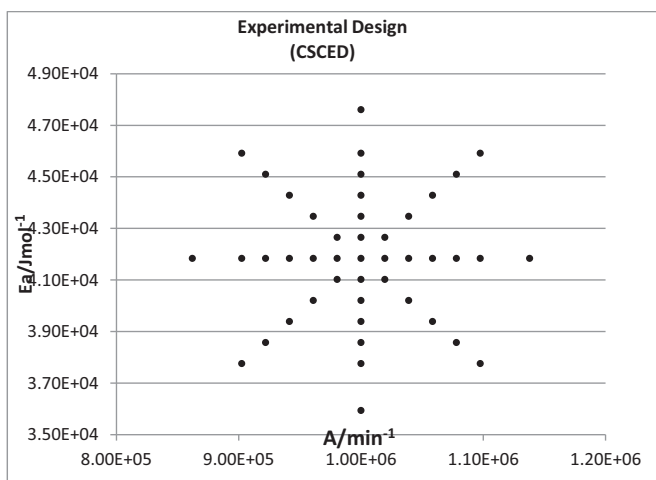


Figure 2. Experimental Design used with a distribution of the experiments corresponding to the named *Central Star Composite Experimental Design (CSCED)*.

3.3. Analysis of the SQD Minimum

When a mathematical optimization algorithm is applied, it is appropriate to analyze graphically *a priori* the profiles of the minima of the SQD, a numerical function dependent upon the ATPs, in order to know whether singular points (*local maxima* and/or *minima*, *saddle points*, etc.) are present in the neighborhood that will make the gradient vector equal to zero, leading the optimization process to fail. It should be noted that to determine the values of ATPs the mathematical expression depicted in equation (12) is fairly complex because it includes several exponential functions with different levels of exponents. Since SQD is a numerical function, we plotted a 3D graphic, generating a large set of non-isothermal curves from pairs of values of the ATPs parameters distributed according to a new CSCED, now in the neighborhood of the minimum that coincides with the *central point* (c_{min}). Later, the $SQD(c)$ values are determined for each curve (c) according equation (17):

$$SQD(c) = \sum_{t=0}^t [\alpha_t(c) - \alpha_t(c_{min})]^2 \quad (17)$$

The α_t values with a hyperbolic T/t variation were generated with the equation (12) from pairs of Ea/A values. There were three matrices corresponding to the three coordinate axes (Ea , A and SQD) whose graphic plot (Figure 3) shows the minimum of the $SQD(c)$ function for both parameters. After the axes had been rotated suitably to obtain the best perspective for visualization, no singular points were observed in the neighborhoods of the minima.

To better confirm this, a “cut” of the 3D figure was made by means of a bisecting plane perpendicular to the XY plane, thereby obtaining a 2D representation (Figure 4) that clearly showed the minimum corresponding to the values of $A= 10^6 \text{ min}^{-1}$ and $Ea= 4.184 \cdot 10^4 \text{ J mol}^{-1}$. It can be observed the absence of any singular points that might lead the optimization process to fail when the AGDC algorithm was applied.

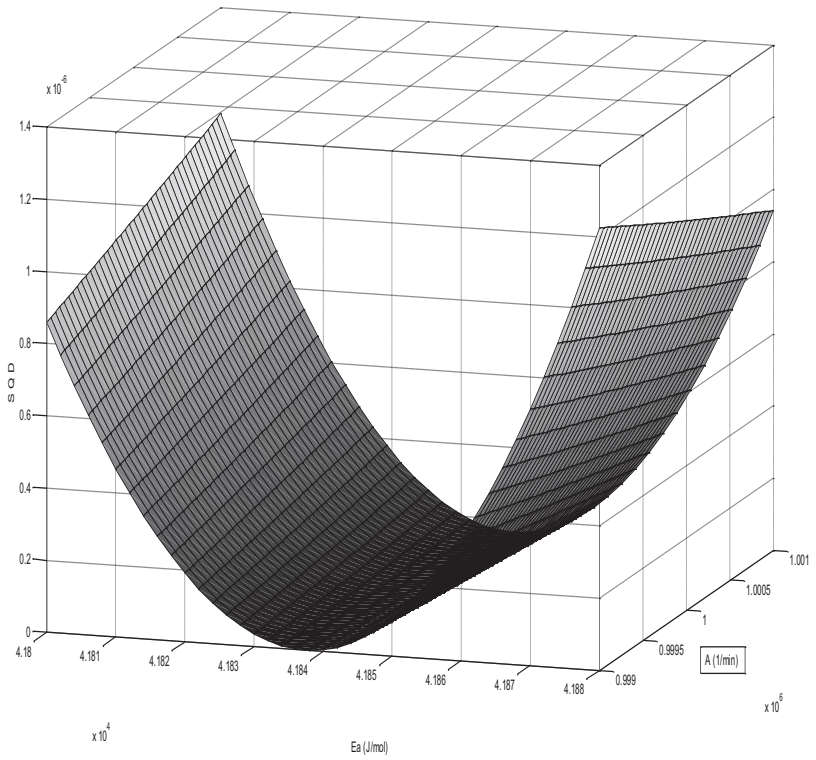


Figure 3. Graphic 3D Plot of the numerical function SQD vs. A and Ea showing the minimum of the function for both parameters.

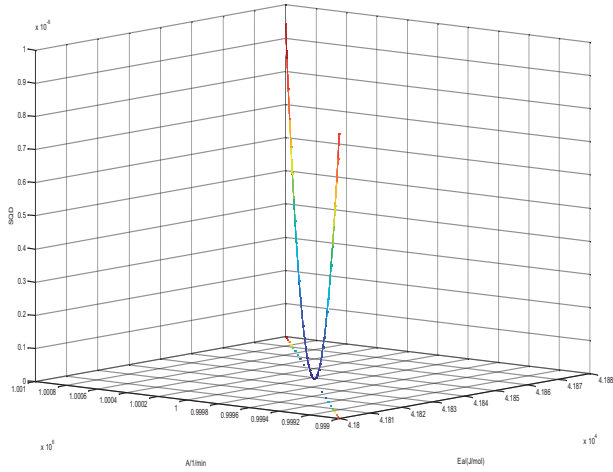


Figure 4. Plot of the “cut” of the 3D figure (Fig. 3) obtained by means of a bisecting plane perpendicular to the XY plane where the minimum with respect to both parameters (A and Ea) is clearly showed.

4. Results and Discussion

ANN treatment involves performing a rigorous process of *training* of the neural network. For this, we carried out a previous study to determine the ideal conditions for application. We generated 45 non-isothermal kinetic curves from eq. (12), considering an inverse hyperbolic variation of T/t according eq. (10), with a value of $m=4.2 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$ in a period of time sufficient for α values of at least 1.5 *half-lives* to be attained. Each curve had 50 data on which random *noise* was imposed with a magnitude of the order of the experimental error of the data ($\pm 1 \cdot 10^{-4}$). For the *back-propagation* algorithm we chose the Levenberg-Marquardt algorithm as the most ideal for performing the optimization of the elements of the *output* and *target* matrices, in all cases using a percentage ratio of 80/10/10 for the *training/validation/testing* processes. To determine the optimum architecture of the neural network, we performed the treatment varying the number and configuration of the *hidden* layer/s systematically, considering all cases possible. Thus, in the case of a neural network (50,1,2) and a variable configuration of the only one *hidden* layer (3-25 neurons), for the processes of *training, validation* and *testing* we obtained *output/target* regression lines with strong dispersions, very poor fitting parameters and unacceptable MSE values, number of *epochs*, μ values, *gradient*, etc.

In the case of the architecture of the neural network with 2 and 3 *hidden* layers (50,2,2) and (50,3,2) and different configurations, we used the same *training* conditions as in the case of 1 *hidden* layer. However, one aspect that merits detailed comment. In the *output/target* regression lines relative to the set of both parameters, the points of the dispersions corresponding to each parameter appear very separated (Figure 5) owing to the great difference in the order of magnitude of E_a (10^4) and A (10^6). In the case of the deviations of A , there is apparently good linearity and in the case of E_a all the dispersion values are superposed and produce a single point, due to the broadness of the scales (10^4 - 10^6) to be considered on both coordinate axes. This strongly masks the true effects of the dispersion, making the statistical fitting parameters of the regression lines obtained for both parameters considered jointly apparently acceptable. However, on considering the *output/target* regression lines of each of the parameters separately, where the appropriate scales are used, a great dispersion is seen in the case of the E_a (Figure 6), corresponding to the dispersion of points observed in Figure 5 values of R^2 (0.1058), slope (0.0886) and ordinate at the origin (0.0383) are obtained that show a very poor correlation between the *output* and *target* values, which make these parameter values determined by ANN unacceptable as final values. Accordingly, it is necessary to further improve the parameter values, which is done by application of the second part of the HA by means of the AGDC gradient method, observing the definitive values of the optimized parameters.

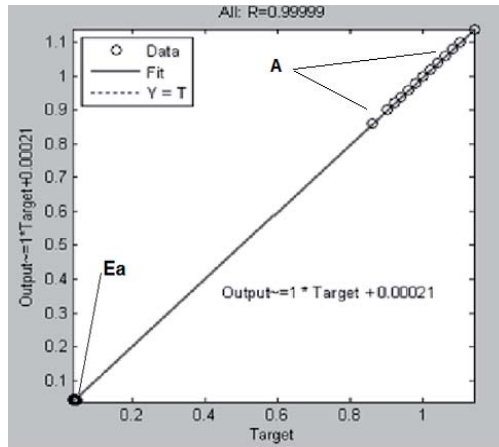


Figure 5. Regression line of the data from the elements of the *Output* vs. *Target* matrices when both parameters (A and E_a) are represented together for the process of *training* of the ANN in the case of the architecture of the neural network with 3 *hidden* layers (50,3,2).

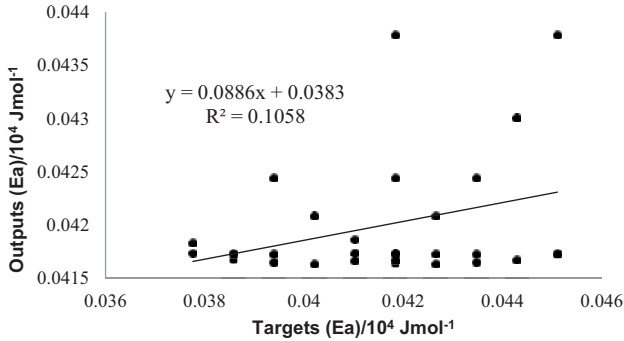


Figure 6. Regression line of the data from the elements of the *Outputs* vs. *Targets* values in the case of exclusively Ea for the process of *training* of the ANN when the architecture of the neural network has 3 *hidden* layers (50,3,2).

It is necessary to quantify the values of these individual *errors* and *Standard Deviations* of each parameter ($SD(A)$ and $SD(Ea)$) obtained with the application of the ANN, with relative *output* and *target* values of the set of 45 curves used in the following general expression for SD :

$$SD(b_{ij}) = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} [(b_{ij}^{Outputs} - b_{ij}^{Targets}) / (b_{ij}^{Targets})]^2}{n_p \cdot n_c} \right)^{1/2} \quad (18)$$

where n_c is the number of curves; n_p is the number of parameters per curve ($n_p=1$) and b_{ij} are the ATPs parameters (Ea and A). The values obtained range between are unacceptable such that it is necessary to improve the results after application of the complete HA, whose results obtained are shown in Table 2.

We performed the *training* of a large set of neural networks considering all the possible configurations for architectures with 2 and 3 *hidden* layers and discarding those of a single layer. In light of the large number of cases assayed, we only show the results of the most significant configurations of the curve corresponding to the central point, because it is the most important one of the 45 comprising the ED. The *output* values of Ea and A and of the deviations in % (*Dev %*) with respect to those that served to generate the data are shown in the first part of Table 2, while the second part shows the values of Ea and A optimized with AGDC together with the deviations in % of the real values (*Dev %*). It is possible to note the clear improvement achieved with application of the complete HA on comparing the values of

the % of deviation (columns 8 and 9) of both parameters, with the corresponding values after the application of ANN (columns 4 and 5).

Config.	ANN				HA = (ANN + AGDC)			
	A /10 ⁶ min ⁻¹	Ea /10 ⁴ Jmol ⁻¹	Dev. % (A)	Dev. % (Ea)	A /10 ⁶ min ⁻¹	Ea /10 ⁴ Jmol ⁻¹	Dev. % (A)	Dev. % (Ea)
14/10	0.9800	4.1933	1.9990	-0.2233	1.0070	4.1859	-0.7038	-0.0453
15/10	0.9801	4.1871	1.9908	-0.0746	0.9955	4.1829	0.4451	0.0253
7/10/10	0.9800	4.1871	1.9965	-0.0738	0.9955	4.1829	0.4548	0.0259
9/10/10	0.9801	4.1950	1.9995	-0.2631	1.0101	4.1867	-1.0104	-0.0640
10/9/10	0.9800	4.1961	1.9995	-0.2885	1.0120	4.1872	-1.2051	-0.0758
10/10/7	0.9804	4.1917	1.9628	-0.1847	1.0043	4.1852	-0.4260	-0.0283
10/10/13	0.9795	4.1938	2.0491	-0.2331	1.0075	4.1860	-0.7521	-0.0483
10/12/10	0.9799	4.1885	2.0002	-0.1082	0.9981	4.1836	0.1880	0.0094
10/20/10	0.9806	4.1893	1.9438	-0.1273	0.9999	4.1841	0.0094	-0.0015
11/10/10	0.9800	4.1923	1.9955	-0.1993	1.0052	4.1854	-0.5212	0.0341
20/10/10	0.9795	4.1935	2.0514	-0.2267	1.0500	4.1866	-0.7017	-0.0452

Table 2. Values of the ATPs, Errors and Deviations obtained after application of ANN and the complete algorithm HA for the *training* processes of a large set of neural networks considering all the possible configurations for architectures with 2 and 3 *hidden* layers

The improvement in the final values of the ATPs is clearly seen on performing a joint plot (Figure 7) of the *Residuals* of α calculated with the values of the parameters obtained by applying the ANN [eq. (19)] and those obtained after the application of the complete algorithm HA [eq. (20)], observing improvements of up to 11-fold (0.023 for HA and 0.25 for ANN) in the time interval of 10-20 minutes.

$$[Res(\alpha)_{ANN} = \alpha_{synt} - \alpha_{cal}(ANN)] \quad (19)$$

$$[Res(\alpha)_{HA} = \alpha_{synt} - \alpha_{cal}(HA)] \quad (20)$$

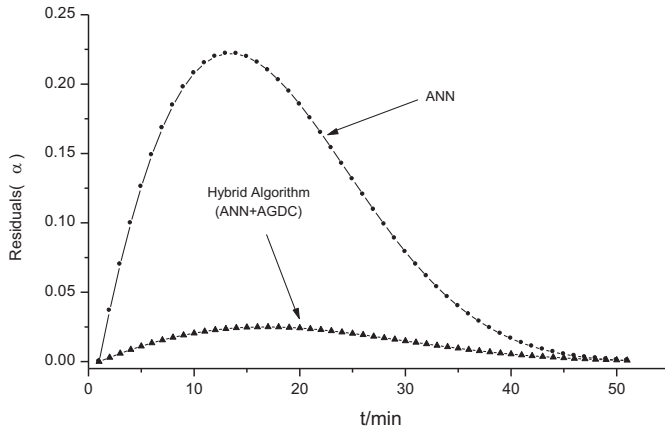


Figure 7. Plot of the *Residuals* of α calculated from equation (19) with the values of the parameters obtained by applying the ANN and those obtained from equation (20) after the application of the complete algorithm HA.

After analyzing the values of all the statistical moments of errors ($SD(A)$ and $SD(Ea)$) and Deviations, it may be concluded that the optimal architecture of the neural network is (50,3,2), constituted by 3 *hidden layers* with a configuration of 10/20/10 used to carry out the later processes of ANN prediction and AGDC optimization.

The process of *prediction* to determine the final optimized parameters consisted of application of the HA to new non-isothermal kinetic curves with “*noise*”, generated from the values of A and Ea within the maximum intervals defined in the ED. For the previous step of *prediction* applying ANN we used the neural network that proved to be optimal after the exhaustive process of *training* -architecture (50,3,2) and configuration (20/10/10)-, in all cases computing a base of 4 kinetic curves, which proved to be the ideal number for carrying out the multiple and simultaneous *prediction* of the initial ANN step. Table 3 shows the results obtained for the *prediction* of 4 pairs of A/Ea values (“*Real Values*”) that served to generate the *input curve base* with “*noise*”. After later application of the AGDC we obtained the optimized values of the parameters (“*Optimized Values*”) resulting from application of the complete HA. Bearing in mind that the percentages of deviations (“% *Dev*”) have an order of magnitude similar to that of the deviations of the *training* process of the neural network, the optimized values of A and Ea can be accepted as being definitive for the 4 *prediction* processes.

HA Algorithm (ANN+AGDC)					
Real Values (A) /10 ⁶ min ⁻¹	Real Values (Ea) /10 ⁴ Jmol ⁻¹	Optimized Values (A) /10 ⁶ min ⁻¹	Optimized Values (Ea) /10 ⁴ Jmol ⁻¹	% Dev. (A)	% Dev. (Ea)
0.980001	4.18914	0.9897	4.1865	-0.99	-0.06
0.960018	4.19443	0.9792	4.1891	-2.00	-0.12
0.980001	4.18914	0.980001	4.1891	-0.00015	-2.11
0.859108	4.18082	0.8537	4.1824	0.61	0.04

Table 3. Values of the ATPs, Errors and Deviations obtained after application of the complete algorithm HA for several *Prediction* processes when a neural network with a optimal architecture (50,3,2) and configuration (20/10/10) has been applied

5. Conclusions

Upon the analysis and comparison of the results shown in Figures 5 and 6 we observed that the graphical representation of the values of targets and outputs jointly corresponding to both parameters (A and Ea) according to the plot obtained by the ANN application of Matlab provides a correlation that seems acceptable. However, on considering the *outputs/targets* regression lines of each of the parameters separately, where the appropriate scales are used, a great dispersion is shown and the results are unacceptable. That justifies the need of later application of the AGDC optimization algorithm in a second step to improve the results as Table 3 and Figure 7 reveals when the *Residuals* from ANN and HA are jointly plotted. The application of the hybrid algorithm (HA) that we propose in the present work provides good and satisfactory results for the optimized values of the ATPs. HA offers several advantages with respect to the isothermal classic methods since it allows to optimize ATPs directly without the need to determine the kinetic constants previously. In addition for the case of non-isothermal experiments, a single replicated kinetic experiment is enough since it allows the computation of a larger number of kinetic data that implies a considerable saving of reagents and laboratory time.

6. References

- [1] J. L. González-Hernández, M. M. Canedo, C. Grande, Optimization of kinetic parameters. Multipurpose KINAGDC(MW) non-linear regression program, *Chemom. Intell. Lab. Syst.* **39** (1997) 77-84.
- [2] M. Maeder, K. J. Molloy, M. M. Schumacher, Analysis of non-isothermal kinetics measurements, *Anal. Chim. Acta* **337** (1997) 73-81.

- [3] A. Niazi, R. Leardi, Genetic algorithms in chemometrics, *J. Chemometrics* **26** (2012) 345-351
- [4] M. Maeder, Y. M. Neuhold, G. Puxty. Application of a genetic algorithm: Near optimal estimation of the rate and equilibrium constants of complex reaction mechanisms, *Chemom. Intell. Lab. Syst.* **70** (2004) 193-203.
- [5] C. Hervás, J. A. Algar, M. Silva, Correction of temperature variations in kinetic-based. Determinations by use of pruning computational neural networks in conjunction with genetic algorithms, *J. Chem. Inf. Comput. Sci.* **40** (2000) 724-731.
- [6] S. Curteanu, H. Cartwright; Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks, *J. Chemometrics* **25** (2011) 527-549.
- [7] F. Amato, J. L. González-Hernández, J. Havel, Artificial neural networks combined with experimental design: A "soft" approach for chemical kinetics, *Talanta* **93** (2012) 72-78.
- [8] J. L. González-Hernández, M. Mar Canedo, S. Encinar, Combining artificial neural networks and experimental design to prediction of kinetic rate constants, *J. Math. Chem.* **51** (2013) 1634-1653.
- [9] M. M. Canedo, J. L. González-Hernández, A new computacional application of the AGDC algorithm for kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* **66** (2003) 63-78.
- [10] M. M. Canedo, J. L. González-Hernández, ANALKIN(AGDC): a multipurpose computational program for the kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* **66** (2003) 93-97.
- [11] M. M. Canedo, J. L. González-Hernández, KINMODEL (AGDC): a multipurpose computational method for kinetic treatment, *J. Math. Chem.* **49** (2011) 163-184.
- [12] M. M. Canedo, J. L. González-Hernández, S. Encinar, Application the computational method KINMODEL(AGDC) to the simultaneous determination of kinetic and analytical parameters, *Appl. Math. Comput.* **219** (2013) 7089-7101 .
- [13] J. L. González-Hernández, M. Mar Canedo, C. Grande-Martín, Combining different mathematical optimization methods: A new "hard-modelling" approach for chemical kinetics, *MATCH Commun. Math. Comput. Chem.* **70** (2013) 951-970.
- [14] K. J. Laidler, A glossary of terms used in chemical kinetics, including reaction dynamics, *Pure Appl. Chem.* **68** (1996) 149-192.
- [15] T. Kohonen, An introduction to neural computing, *Neural Networks* **1** (1988) 3-16.
- [16] MathWorks MatLab, R2012a, Vs7.14.0.739 (2012).
- [17] M. A. Wolfe, *Numerical Methods for Unconstrained Optimization*, Van Nostrand, Berkshire, 1978.

- [18] M. M. Canedo, J. L. González-Hernández, ANALKIN(AGDC): a multipurpose computational program for the kinetic resolution of multicomponent mixtures (static and dynamic), *Chemom. Intell. Lab. Syst.* **66** (2003) 93-97.
- [19] M. M. Canedo, J. L. González-Hernández, KINMODEL (AGDC): a multipurpose computational method for kinetic treatment, *J. Math. Chem.* **49** (2011) 163-184.
- [20] P. Gill, W. Murray, M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.
- [21]. M. Meloun, J. Havel, E. Hölfeldt, *Computation on Solution Equilibria*, Horwood, Chichester, 1988.

A robust hybrid algorithm (neural networks-AGDC) applied to non-isothermal kinetics of consecutive chemical reactions

Sonsoles Encinar · J. L. González-Hernández ·
M. Mar Canedo · Diana Juanes

Received: 26 October 2014 / Accepted: 5 January 2015 / Published online: 14 January 2015
© Springer International Publishing Switzerland 2015

Abstract This paper is concerned with the application of a Hybrid Algorithm (HA) to the determination of the Thermodynamic Activation Parameters (ATP) of a kinetic system of first order consecutive reactions. The 8 ATP's parameters involved in the Arrhenius and Eyring equations have been directly determined from the non-isothermal kinetic data without prior knowledge of the rate constants. AH is constituted by a combination of two algorithms based on different mathematical principles which are sequentially applied. In a first step, a “*soft modeling*” method of Artificial Neural Networks (ANN) is applied and the obtained values of ATP's parameters are used as *initial estimates* of a new optimization algorithm (AGDC) applied in a second stage to improve the values of the final parameters. The great success of HA is the efficient resolution of the ambiguity of the results obtained by ANN. In addition, comparing with the classic algorithms, which present the known weak points, HA offers important advantages: (a) the lack of necessity to know a priori the *initial estimates* since they are calculated from ANN application, (b) the low probability of being trapped at local minima, saddle points, etc. by means of the exhaustive control and suitable correction of the movement vector during the optimization process, and (c) the simultaneous determination of a higher number of parameters endowed with very different orders of magnitude.

Keywords Hybrid algorithm · Neural networks · AGDC · Non-isothermal kinetics · Thermodynamic activation parameters

S. Encinar · J. L. González-Hernández (✉) · M. M. Canedo · D. Juanes
Department of Physical Chemistry, Faculty of Chemistry, University of Salamanca,
37008 Salamanca, Spain
e-mail: jlgh93@usal.es

1 Introduction

The classic method to obtain the values of the Activation Thermodynamic Parameters (ATP) of a chemical reaction consists in the previous determination of the pairs of values of the rate constant $k(T)$ from the isothermal kinetic experiences at different temperatures. Posteriorly, all pairs are fitted to straight lines by using Linear Regression, according the linearized Arrhenius and Eyring equations [1]. From each individual $k(T)$ of each single reaction, we determine the values of the pre-exponential Factor (A) and the Activation Energy (E) from Arrhenius equation and the Activation Enthalpy (ΔH^\ddagger) and the Activation Entropy (ΔS^\ddagger) from the Eyring's equation, obtained from the Transition-State Theory (TST).

As for kinetic systems formed by “r” reactions with $k_r(T)$ rate constants, the procedure consists in the application of hard modeling methods in order to set up the system of ordinary differential equations (ODE). After solving them, the explicit mathematical expressions of the concentrations of all species involved in the kinetic system can be determined on the basis of $k_r(T)$ and time [2]. By using experimental data from a kinetic isothermal set carried out to different temperatures, the values of $k_r(T)$ can be calculated fitting the data to these explicit functions. Finally, the ATP's values are determined from the pairs of values k_r/T . This method leads to significant drawbacks: (a) a calculation of a high number of parameters is required, as it happens in our case of the consecutive reactions, where 8 ATP parameters have to be determined, (b) frequently, the ODE system doesn't provide exact mathematical solutions, so the $k_r(T)$ cannot be directly obtained and, therefore, ATP's values also remain unknown. Approaches (i.e. *Steady State* method) can be used to determine the values of the rate constants but the final ATP's values are not accurate at all, (c) a big amount of experimental work at the laboratory is needed to determine a reduced collective of value pairs $k_r(T)/T$, (d) it is also very common the necessity of knowing the values of other non kinetic magnitudes (i.e. equilibrium constants, etc.), which must be previously obtained by performing an additional task.

As a result of the foregoing, it is more convenient to design a new method to find directly the ATP's values without a previous determination of the rate constants. This new one requires to acquire a large amount of experimental information, as the data from the non-isothermal kinetics, which provide a more numerous set of data and final results more accurate. Therefore, in order to obtain the ATP' values, it is necessary to design a new algorithm of computational treatment, as the one proposed on the present work.

The computational determination of the parameters involved in mathematical functions is usually carried out by means of mathematical optimization methods applying numeric second order gradient algorithms [3]. These non-linear iterative fitting techniques are particularly used in Chemistry as well as other fields of Science. However, one of the most drawbacks of these optimization methods is their high sensitivity to the initial estimates of the supplied parameters. Only if these values are very close to the global minimum can a fast and reliable convergence of the iterative process be expected, guaranteeing success of the parameter optimization. If the initial estimates are far from the global minimum the process may become divergent or reach a singular point (“local minimum”, “saddle point”, etc.), leading the optimization process to fail.

This often happens in the treatment of kinetic models when the number of parameters to be optimized is moderately high. Additionally there is a series of problems involving identifiability and distinguishability [4,5], which lead to different types of ambiguity in the solutions to the stiff systems of ordinary differential equations (ODE). In light of this, it would seem appropriate to design and apply a new method that, initially, would provide an approach to the global optimum and then use such results as a starting point to apply a robust gradient method that will guarantee the success of the mathematical optimization of parameters.

In the present work we designed and successfully applied to the kinetic system of consecutive reactions, a new Hybrid Algorithm (HA) able to determine the ATP's values directly from non-isothermal kinetic data without the need of determining the rate constants in a previous step.

A HA is the one in which 2 or more different algorithms are combined in order to solve the same mathematical problem. Depending on data, the HA can perform by means of choosing the algorithm that fits better the data or switching between them during the whole application process. This second way corresponds to the HA proposed in this work. The final objective is to reach the combination of the most appropriate characteristics of each one, so the general overall HA can perform better than individual components. The HA refers not only to a combination of multiple algorithms to solve a different problem. In fact, many algorithms can be considered themselves as combinations of simple pieces. HA is a combination of algorithms that solve the same problem but differ in their characteristics, particularly in efficiency.

The HA proposed in this work comprises two algorithm based on different mathematical principles in which sequential application is carried out. A “soft modeling” method is applied by using Artificial Neural Networks (ANN) [6], which presents the valuable advantage of making the use of initial estimates of the parameters unnecessary. The final objective is to determine the parameter values (outputs) in the neighborhood of the optimum global. These values have been used as the initial estimates values of a robust and efficient numeric second order gradient algorithm (AGDC) [7], able to reach the desired global minimum to guarantee the success of the final optimization of the parameters. The application of HA to the kinetic system of consecutive reactions provides several important advantages since the direct optimization of those parameters became possible, with no need of calculating the rate constants $k_r(T)$ in previous steps. It must be considered that 8 ATP parameters are involved in the studied kinetic system of consecutive reactions, representing a very elevated number of parameters to be determined. The treatment requires to acquire the kinetic data from non-isothermal kinetic experiments, imposing a controlled variation of temperature along the reaction kinetic. The number of the set of kinetic data computed in the classic isothermal procedure is very small since it is limited by experimental conditions, when dealing with a low number of isothermal experiments (normally, 10–12), each of them performed at a different temperature. In the case of non-isothermal experiments, a single replicate kinetic experiment is enough since it allowed the computation of a huge set of kinetic data which, bearing in mind the laboratory time and the amount of reagents saved, is a great and important advantage.

The literature contains several recent references of the AGDC mathematical optimization algorithm that is a symbolic second-order gradient method that performs a rigorous analysis and control of the movement vector and of each of its terms. The numeric version of this algorithm was designed and successfully implemented at our laboratory within the field of Chemistry in the treatment of many systems with kinetic, analytical and thermodynamic purposes. Thus computational applications have developed for the quantitative analysis of static (SMM) and dynamic (DMM) multicomponent mixtures [7, 8] or for the determination of kinetic constants in different reaction mechanisms after the application of the numerical computational program KINMODEL(AGDC) [9]. In this type of joint optimization of several parameters the problem of ambiguity in the solution is analyzed in reaction mechanisms comprising first-order reactions since it is common to find several (two or more) groups of parameters that fit the experimental kinetic data. The AGDC algorithm has also been used for the determination of the initial concentrations of the species involved in the reaction mechanism, individually or jointly, with the kinetic constants of the elementary reactions comprising the mechanism [10]. In this paper there is analyzed the influence of a series of factors affecting the optimization process and in addition there is studied the possibility of the existence of ambiguity in the solutions, since parameters of different natures and orders of magnitude are determined. Recently a novel “*hard-modeling*” computational approach, based in the algorithm AGDC(MW) [11], is proposed for the simultaneous optimization of the rate constants and molar absorption coefficients. The approach is the result of combining 3 different strategies based in 3 versions of AGDC and an evident improvement in the values of the optimized parameters has been obtained in comparison with the application of a single strategy. Several authors have applied Hybrid Algorithms (HA) in the field of Chemistry in the literature that essentially comprises Genetic Algorithms (GA). Their design and application are currently able to solve problems pending solution and reveal the high degree of reliability and precision in the results obtained. Of interest is one review [12] in which the authors consider a broad range of applications of different types of GA in Chemometrics: Maeder et al. [13] determine the rate and equilibrium constants of reaction mechanisms by application of a HA based on a GA. Hervás et al. [14] use GA and pruning computational neural networks for selecting the number of inputs required to correct temperature variations in kinetic-based determinations. Artificial Neural Networks offer a versatile “*soft modeling*” method that can be applied in diverse fields with acceptable results [15]. The method is applied for quantitative purposes, among others, in the so-called *Principal Component Analysis* (PCA) method, in which there are no explicit functions of multivariate correlation or, if there are, they are extraordinarily complex. The application of ANN (alone or combined with a mathematical optimization algorithm) in chemical kinetics when a single chemical reaction is considered, [16–18] can provide acceptable results. However, it is necessary to perform an exhaustive process of *training* of the neural network in order to obtain its optimal architectures, meaning that the method is time-consuming and tedious and that it cannot be used individually in all cases. Nevertheless, it is an ideal method for carrying out the approximation of the ATP’s values to those of the global minimum, without previously knowing anything about the magnitude and sign of the parameters.

2 Theoretical aspects

2.1 Kinetic and thermodynamic aspects in non-isothermal kinetics

Let us consider in general a chemical system formed by n_r chemical elementary reactions where n_s chemical species can be involved. According to IUPAC's norms [19], the r -th chemical reaction can be expressed for the generic equation

$$0 = \sum_{j=1}^{n_s} \nu_{j,r} B_j \quad (1)$$

where, B_j represents the chemical species involved in the system of reactions; $r = (1, \dots, n_r)$, the number of chemical reactions; $j = (1, \dots, n_s)$, the number of chemical species; $\nu_{j,r}$, the stoichiometric coefficient of the species B_j in the r -th reaction; $\nu_{j,r} < 0$ when B_j plays only the role of reactant in the r -th reaction and $\nu_{j,r} > 0$ when B_j plays only the role of product in the r -th.

When the reaction is an elementary or concerted one, the absolute values of the kinetic order ($z_{l,r}$) and stoichiometric coefficient of B_j coincide, that is $|\nu_{l,r}| = |z_{l,r}|$. The rate differential equation of the chemical species B_j in the r -th is given by

$$d[B_j]/dt = k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \quad (2)$$

where B_l are the species playing only the role of reactants in the r -th reaction ($\nu_{l,r} < 0$) and k_r is the kinetic rate constant of the r -th reaction. Each chemical species can take part in several reactions and the rate differential equations will be the sum extended over those reactions where the reactant B_l appears, obtaining a system of ordinary differential equations (ODE) according to the generic equation,

$$d[B_j]/dt = \sum_{r=1}^{n_r} k_r \nu_{j,r} \prod_{l=1}^{n_s} [B_l]^{|z_{l,r}|} \quad (3)$$

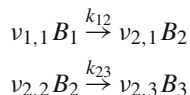
The general solution of the system of rate ODE give the explicit function of the concentrations of the all species with time ($[B_j]_{t_i}$). If the experimental data are expressed in absorbance, we have to consider the Lambert–Beer–Bouguer law:

$$A_{j,t_i}^\lambda = \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad (4)$$

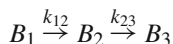
where A_{j,t_i}^λ is the absorbance of the species B_j at the, time t_i and path length 1 cm and ε_j^λ is the molar absorption coefficient of B_j at the wavelength λ . The absorbance of the mixture (A_{T,t_i}^λ) measured at wavelength λ , time t_i and temperature T and, can be expressed as:

$$A_{T,t_i}^\lambda = \sum_{j=1}^{n_s} A_{j,t_i}^\lambda = \sum_{j=1}^{n_s} \varepsilon_j^\lambda \cdot [B_j]_{t_i} \quad (5)$$

The chemical system formed by 2 first order consecutive reactions has been studied in the present work and can be represented according to Eq. (1), as



Considering $\nu_{1,1} = \nu_{2,2} = -1$ and $\nu_{2,1} = \nu_{2,3} = 1$, we have the first order consecutive reactions system studied in this work.



According this, the system of ordinary differential equations (ODE) can be expressed using matrix notation as

$$d/dt \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} = \begin{bmatrix} -k_{12}(T) & 0 & 0 \\ k_{12}(T) & -k_{23}(T) & 0 \\ 0 & k_{23}(T) & 0 \end{bmatrix} \begin{bmatrix} [B_1] \\ [B_2] \\ [B_3] \end{bmatrix} \quad (6)$$

2.1.1 Solution of the first differential equation of the ODE system

The first rate differential equation of the ODE system (6) is

$$d[B_1]/dt = -k_{12}(T) [B_1] \quad (7)$$

that can be individually solved taken in account that only depends of $[B_1]$. It is convenient to consider here the general kinetic treatment of a chemical reaction in non isothermal conditions.



The rate differential equation to this reaction, expressed in function of the *extent of reaction* variable in units of molar concentration (ξ_1') when the reactant B_1 obey to a first order kinetic, can be written as:

$$d\xi_1'/dt = k_{12}(T) ([B_1]_0 - |\nu_1| \xi_1') \quad (9)$$

where $[B_1]_0$ is the initial concentration of B_1 species and $k_{12}(T)$ is the rate constant, a function of T in non-isothermal conditions. Separating variables and after integration of the first member of the equation, we have,

$$- (1/|\nu_1|) \ln \{ ([B_1]_0 - |\nu_1| \xi_1') / [B_1]_0 \} = \int_0^t k_{12}(T) dt \quad (10)$$

The Activation Energy, (E_{12}) and the pre-exponential Factor (A_{12}) can be determined substituting $k_{12}(T)$ in Eq. (10) according the Arrhenius equation, where α_1 is the remaining molar fraction of the reactant B_1 .

$$-(1/|v_1|) \ln \alpha_1 = \int_0^t A_{12} e^{-E_{12}/RT} dt \quad (11)$$

In non-isothermal conditions, the second member of Eq. (11) cannot be integrated since there are two dependent variables [$T = f(t)$]. It will be crucial to establish the identity of this function because the mathematical method of resolution of the Eq. (11) will be different. However 2 requirements must be met: the function must be monotonically increasing, since is this way it will be possible to minimize the great differences in the reaction rate existing between the beginning and the end of the reaction; the rate of heating must be suitable for the interval of time studied and its profile must be reproducible in the laboratory. In order to choose a variation type T/t we consider two possibilities for the function:

1. hyperbolic branch

$$1/T = (1/T_0) - mt \quad (12)$$

which leads to the following equation:

$$-(1/|v_1|) \ln \alpha_1 = \int_0^t A_{12} e^{-E_{12}/R[(1/T_0)-mt]} dt \quad (13)$$

This hyperbolic function has *a priori* a great advantage since the primitive function of the integral of the Eq. (13) exists being the mathematical exact solution the following expression:

$$\alpha_1 = e^{-[|v_1|A_{12}R/mE_{12}](e^{-E_{12}/RT_0})(e^{mE_{12}/Rt} - 1)} \quad (14)$$

However 2 considerations must be done: (i) the explicit function of α_1 depending on the time (14) is not simple and there is necessary the application of a method of sufficiently robust treatment for the determination of the Thermodynamic parameters E_{12} and A_{12} . In addition it is necessary to consider the great difference in the order of magnitude of both parameters that complicates extraordinarily the success in the application of the method of treatment. (ii) the experimental points (T/t) must satisfy to this hyperbolic function and therefore, it is necessary to reproduce the profile of the curve of the function (T/t) (12) with the points obtained in the laboratory.

2. function of polynomial type of n -th degree

$$T = \sum_{j=0}^{j=n} a_j t^j \quad (15)$$

or even, we can propose a first order polynomial, that is, a lineal function ($T = T_0 + a_0 t$)

$$-(1/|v_1|) \ln \alpha_1 = \int_0^t A_{12} e^{-E_{12}/R(1/\sum_{j=0}^{j=n} a_j t^j)} dt \quad (16)$$

This equation does not have mathematical exact solution since the integration cannot be performed. We have 2 options of treatment: (a) performing a numerical integration of the Eq. (13) using appropriate *quadrature* formulas applying a suitable numerical algorithm for the resolution of integrals (Simpson, Lobatto, Gauss-Kronrod, Vectorized, etc.), and (b) numerical resolution directly from the beginning of the ordinary differential equation (7) expressed in terms of Arrhenius's equation, that is:

$$d[B_1] = -[B_1] A_{12} e^{-E_{12}/RT} dt \quad (17)$$

The solution of this differential equation can be performed by means of the application of numerical methods that they must be suitable for the treatment of *stiff* systems (Runge–Kutta methods, Gear's method (BDFs), Rosenbrock formula, trapezoidal rule with "free" interpolant...etc).

In the case of the determination of the ATP's parameters involved in the Eyring's equation (ΔH_{12}^\ddagger and ΔS_{12}^\ddagger) the mathematical treatment of the differential rate Eq. (10) is different with respect to the case of Arrhenius. It can be expressed as:

$$-(1/|v_1|) \ln \alpha_1 = \int_0^t (k_B/h) T e^{-\Delta H_{12}^\ddagger/RT} e^{\Delta S_{12}^\ddagger/R} dt \quad (18)$$

The solution of the integral in the second member of (18) depends of the type of the explicit function $T = f(t)$ used. We discard the inverse function of the temperature corresponding to a hyperbolic branch (12) because in this case is not possible a mathematical exact solution of the integral. Therefore, we consider exclusively the polynomial function of n-th degree (15) and the treatment to be performed is identical than the case of Arrhenius's equation. We consider 2 options: numerical integration of the equation using appropriate *quadrature formulas* and the numerical resolution directly from the beginning of the (ODE) (18), by means of the application of methods suitable for the treatment of stiff systems.

$$d[B_1] = -[B_1] (k_B/h) T e^{-\Delta H_{12}^\ddagger/RT} e^{\Delta S_{12}^\ddagger/R} dt \quad (19)$$

2.1.2 Solution of the other ODE system differential equations

The remaining differential equations that constitute the system (6) for the substances B_2 and B_3 , are:

$$d[B_2]/dt = k_{12}(T)[B_1] - k_{23}(T)[B_2] \quad (20)$$

$$d[B_3]/dt = k_{23}(T)[B_2] \quad (21)$$

which resolution in isothermal conditions lead to the solutions:

$$[B_2] = \frac{[B_1]_0 k_{12}}{k_{23} - k_{12}} \left(e^{-k_{12}t} - e^{-k_{23}t} \right) \quad (22)$$

$$[B_3] = [B_1]_0 \left(1 - \frac{k_{23}}{k_{23} - k_{12}} e^{-k_{12}t} + \frac{k_{12}}{k_{23} - k_{12}} e^{-k_{23}t} \right) \quad (23)$$

However, if non-isothermal conditions are considered and after substituting $k_{12}(T)$ and $k_{23}(T)$ on Eqs. (20) and (21) based on the Arrhenius and Eyring's functions, these differential equations do not provide us an exact mathematic solution. In conclusion, there is non-explicit functions of $\alpha_2(t_i)$ and $\alpha_3(t_i)$ depending of ATP parameters and time. The solutions are exclusively numeric ones, what it means only discrete values of $\alpha_2(t_i)$ and $\alpha_3(t_i)$ can be determined for each time value. For that reason, it is necessary to determine $k_{12}(T_i)$ y $k_{23}(T_i)$ for each value of T_i , by substituting the t_i values on the following equations:

$$\begin{aligned} k_{12}(T_i) &= A_{12} e^{-E_{12}/R \sum_{j=0}^{j=n} a_j t_i^j} \\ &= (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{12}^\ddagger/R \sum_{j=0}^{j=n} a_j t_i^j} e^{\Delta S_{12}^\ddagger/R} \end{aligned} \quad (24)$$

$$\begin{aligned} k_{23}(T_i) &= A_{23} e^{-E_{23}/R \sum_{j=0}^{j=n} a_j t_i^j} \\ &= (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{23}^\ddagger/R \sum_{j=0}^{j=n} a_j t_i^j} e^{\Delta S_{23}^\ddagger/R} \end{aligned} \quad (25)$$

It is important to consider that the values of A_{12} , E_{12} , ΔH_{12}^\ddagger and ΔS_{12}^\ddagger are previously known from the application of ANN method. These values are involved in Eq. (24) and they must be constant in the later process of determination and optimization of A_{23} , E_{23} , ΔH_{23}^\ddagger and ΔS_{23}^\ddagger by means of complete application of HA to Eq. (25). Finally the numeric values of $\alpha_2(t_i)$ y $\alpha_3(t_i)$ are determined by the Eqs. (22) and (23).

$$\alpha_2(t_i) = \frac{k_{12}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} \left(e^{-k_{12}(T_i)t_i} - e^{-k_{23}(T_i)t_i} \right) \quad (26)$$

$$\alpha_3(t_i) = \left(1 - \frac{k_{23}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} e^{-k_{12}(T_i)t_i} + \frac{k_{12}(T_i)}{k_{23}(T_i) - k_{12}(T_i)} e^{-k_{23}(T_i)t_i} \right) \quad (27)$$

These 2 sets of kinetic value pairs $[\alpha_2(t_i)/t_i]$ and $[\alpha_3(t_i)/t_i]$ are going to constitute the data of the *input curves* for their treatment with the HA, which it will provide us all the ATP's values A_{12} , A_{23} , E_{12} , E_{23} , ΔH_{12}^\ddagger , ΔH_{23}^\ddagger , ΔS_{12}^\ddagger and ΔS_{23}^\ddagger .

2.2 Hybrid Algorithm: HA (ANN-AGDC)

The application of *Hybrid Algorithm (HA)* to kinetic system of consecutive reactions has been carried out by 2 different procedures, depending if it is implemented into the

resolution of the 1st differential equation (7) from the 3 equations of the ODE system (6), or into the resolution of the other 2 differential equations of the system:

I. Resolution of the 1st differential equation of the ODE system (6).

The treatment of α_1/t data pairs is accomplished in order to determine the ATP parameters's A_{12} , E_{12} , ΔH_{12}^\ddagger and ΔS_{12}^\ddagger , by following 2 different steps:

(1) **Method ANN.** The input data are constituted by a matrix of curves (α_1/t) of synthetic kinetic data endowed with “noise” and 2 *targets* matrices, which contain the ATP's values for each curve consistently distributed on the basis of the appropriate *Experimental Design*. The results are 2 *outputs* matrices composed by the ANN obtained values of the former 4 ATP parameters, which will be the *initial estimates* for the following step.

(2) **AGDC algorithm.** The values of the 4 ATP parameters are optimized by implementing the numeric algorithm AGDC adapted to the considered cases and depending on the type of T/t function chosen. The matrix of curves (α_1/t) is used as *input data* and the 4 ATP parameters, determined by the previous ANN method application, as *initial estimates*. The final solutions obtained by the HA application are the optimized values of the 4 parameters A_{12} , E_{12} , ΔH_{12}^\ddagger and ΔS_{12}^\ddagger .

II. Resolution of the other 2 differential equations of the ODE system

The treatment of α_2/t and α_3/t data pairs is performed in order to determine all the ATP parameters. The whole procedure comprises 2 steps:

(1) **Method ANN.** The *input data* is constituted by the curves' matrix of the kinetic pairs α_2/t and α_3/t of synthetic data with *noise* and the 2 *targets* matrices constituted by the values of ATP's A_{12} , E_{12} , A_{23} , E_{23} (Arrhenius) and ΔH_{12}^\ddagger , ΔS_{12}^\ddagger , ΔH_{23}^\ddagger , ΔS_{23}^\ddagger (Eyring), respectively. These 2 matrices contain 2 types of ATP parameters: (a) the ones with subscripts (12), which values were optimized by the application results from procedure I and will remain constant and invariable (24) along the steps of the procedure II. (b) the ones with subscripts (23), which values will be conveniently modified—Eq. (25) for the determination from ANN and the optimization by AGDC. The solutions are 2 *outputs* matrices that contain the values of ANN determined from the 2 sets of 4 ATP Arrhenius and Eyring parameters (8 in total) and will be the *initial estimates* for the following step of the process.

(2) **AGDC algorithm.** The numeric algorithm AGDC, is applied depending on the kind of T/t function in order to optimize the 4 ATP parameters with subscripts (23), that is, A_{23} , E_{23} , ΔH_{23}^\ddagger , ΔS_{23}^\ddagger . By way of explanation, the AGDC iterative process starts up with the *initial estimates* values of the parameters. Although the 4 ATP parameters with subscripts (12) are not modified, the other 4 ATP ones with subscripts (23) are determined and optimized by the ANN and AGDC sequential processes. The final solutions after application of the HA are the values of 8 ATP parameters (A_{12} , A_{23} , E_{12} , E_{23} , ΔH_{12}^\ddagger , ΔH_{23}^\ddagger , ΔS_{12}^\ddagger and ΔS_{23}^\ddagger).

2.2.1 Artificial neural networks

Artificial Neural Networks are parallel interconnected networks of simple computational elements called neurons and are structured in layers that are intended to interact with the objects of the real world in a similar way to the biological nervous systems

[20]. Parallel processing is the ability of the brain to simultaneously process incoming stimuli of differing quality. The multilayer neural network uses sets of input data and parameters (called *targets*), distributed in 2 input matrices when Matlab [21] is applied. The elements of the input matrix are the calculated synthetic values, where one row contains a single curve of the data and all the curves thus obtained (n_c) are grouped in an *input data* matrix. The *target matrix* is formed by the sets of parameters (n_p). In our case, the input data matrix contained the kinetic data of all curves ($\alpha_j, A_T, A_j, [B_j]$) and the *target matrix* ($n_c \times n_p$) contained the set of kinetic rate constants (k_{mn}). Formally, a multilayer neural network is an oriented graph in which the nodes represent a set of processing units, called neurons, and the connections represent the information flow channels. Each connection between two neurons has an associated value called *weight* which specifies the strength of the connection between neurons. Positive and negative values determine excitatory and inhibitory connections, respectively. The choice of a specific class of networks for the approximation of a nonlinear map depends on a variety of factors dictated by the context and is related to the desired accuracy and the prior information available concerning the input–output pairs.

The first layer of a multilayer neural network contains neurons that receive the input data values from the elements of the input data matrices. This information is transmitted from the i -th neuron of a layer to the j -th neuron of the subsequent one, with a weight w_{ji} . A neuron parameter (*bias*) is summed with the weighted inputs of the neurons and passed through the transfer function to generate the output of the neurons.

The layer following the input one is called *hidden*. In each neuron of a *hidden* layer the weighed inputs coming from the previous one are summed with each other and added to a *bias*. The result is then transformed by means of a suitable mathematical function to obtain an output called *activation of the neuron*, which is transferred to the neurons in the next layer after another weighing step. The output parameters values are calculated in the last layer (*output* layer) by means of a suitable transformation function.

The process described is called to as the *training or learning* of the multilayer neural network and constitutes an iterative method where the iterations are called *epochs*. After each *epoch*, the calculated values of the parameters are grouped in the *output matrix* (b_{ij}^{output}) and they are compared with those of the corresponding curve in the *target matrix* (b_{ij}^{target}) and the optimum value of the Mean Squared Error (MSE), expressed in absolute value, is calculated according Eq. (28)

$$MSE = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} (b_{ij}^{output} - b_{ij}^{target})^2}{n_p \cdot n_c} \right)^{1/2} \quad (28)$$

where n_c is the input number of curves and n_p is the number of parameters, $n_c \times n_p$ being the dimensions of both matrices (*output matrix* and *target matrix*).

During the process of *training*, *weights* and *bias* values are modified with suitable mathematical optimization algorithms in order to minimize the calculated values of MSE in each *epoch*. In the present work, the *back-propagation* algorithm was used.

The iterative process finishes when the minimum value of MSE is reached, after which the *training* process can be considered to be completed.

It is necessary to know the optimal architecture and topology of the multilayer neural network in order to obtain the best results when ANN is applied to the system under study. We have used a method of *trial and error* by minimizing the MSE values obtained for the different possible configurations of the same number of *hidden* layer/s. It must to determine the minimum value (optimum) of the MSE for all possible configurations (i.e. in the range between 1 and 12) for the *hidden* layer/s chosen. For each *hidden* layer, a graph of MSE values versus the number of neurons shows that initially, for the lower configurations, the value of the MSE decreases rapidly when the number of neurons increases, but after a constant value or a poor improvement is obtained. The optimum number of neurons (*configuration*) in that *hidden* layer is given by the point of intersection of the two branches of the graph. Sometimes, a small minimum appears near this intersection point. The architecture of the neural network can be written in abbreviated notation as $(n_{inp}, n_{hid}, n_{out})$, where n_{inp} is the number of neurons in the *input* layer, n_{hid} in the *hidden* layer and n_{out} in the *output* layer.

Neural network *training* is completed with the processes of *validation* and *testing* reach satisfactory results. These are 2 control and verification processes of the iterative minimization method between the elements of the *output* and *target* matrices. Among the different curves comprising the *input* matrix, random choice is made of a percentage of the total, established previously (5, 10% . . .), which gives rise to a “*sub-matrix*” of input curves that are subjected to iterative optimization until a minimum MSE value is reached. It is thus possible to verify the validity of the *training* process by ensuring that it is convergent, that it has an appropriate termination, and that there not been any *overfitting*, since any possible *overtraining* has taken been into account. *Validation* process is completed when in a given number (≥ 6) of consecutive *epochs* the MSE remains constant or shows a slight tendency to increase. The *testing* process is similar, except that the control for terminating the process is performed by controlling the computation time instead of the number of *epochs*. The process of *prediction* consists of the determination of the unknown parameters from a set of experimental data after application of the optimal and trained neural network. Obviously, the elements of the *target* matrix are unknown for this *prediction* process, and only the *input data* matrix is provided to the neural network. The elements of the *input data* matrix in the process of *prediction* will be experimental kinetic values (α_j , A_T , A_j , $[B_j]$, etc.) acquired from a kinetic system of reactions developed at the laboratory.

2.2.2 AGDC algorithm

The AGDC is an algorithm of mathematical optimization based on a second-order gradient method that minimizes, by means of an iterative process, the numerical function SQD:

$$\text{SQD}(\mathbf{X}) = \sum_{i=1}^{N_d} ((\alpha_j(t_i))_C - (\alpha_j(t_i))_E)^2 \quad (29)$$

$$\alpha_j(t_i) = [B_j]_i / [B_1]_0 \quad (30)$$

where \mathbf{X} is the vector that contains the parameters to be optimized. In this case the components of the \mathbf{X} vector can be pairs of values of the Activation Thermodynamic parameters involved in a first order consecutive reactions system (E_{12} and A_{12} , ΔS_{12}^\ddagger and ΔH_{12}^\ddagger , E_{23} and A_{23} , ΔS_{23}^\ddagger and ΔH_{23}^\ddagger). The AGDC algorithm first uses, as the movement vector, the one indicated by the Gauss–Newton method [3, 22]:

$$\mathbf{p}^{(m)} = -\mathbf{g}^{(m)} [\mathbf{H}^{(m)}]^{-1} \quad (31)$$

where $\mathbf{p}^{(m)}$, $\mathbf{g}^{(m)}$ and $[\mathbf{H}^{(m)}]^{-1}$ are respectively the movement vector, the gradient vector and the inverse of the Hessian matrix of the iteration m , whose terms are derived from the function to be minimized (SQD) with respect to each of the parameters to be determined (\mathbf{X}).

If the residuals are given by:

$$RES_i = (\alpha_j)_C - (\alpha_j)_E \quad (32)$$

then $\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$ are given by:

$$\mathbf{g} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} RES_i \frac{\partial(\alpha_j)_C}{\partial X_1} \\ \sum_{i=1}^{N_d} RES_i \frac{\partial(\alpha_j)_C}{\partial X_2} \end{bmatrix} \quad (33)$$

$$\mathbf{H} = 2 \begin{bmatrix} \sum_{i=1}^{N_d} \left(\frac{\partial(\alpha_j)_C}{\partial X_1} \right)^2 & \sum_{i=1}^{N_d} \left(\frac{\partial(\alpha_j)_C}{\partial X_1} \right) \left(\frac{\partial(\alpha_j)_C}{\partial X_2} \right) \\ \sum_{i=1}^{N_d} \left(\frac{\partial(\alpha_j)_C}{\partial X_2} \right) \left(\frac{\partial(\alpha_j)_C}{\partial X_1} \right) & \sum_{i=1}^{N_d} \left(\frac{\partial(\alpha_j)_C}{\partial X_2} \right)^2 \end{bmatrix} \quad (34)$$

Calculation of the derivatives is performed numerically by means of the central differences method [23] and the inverse of the Hessian matrix is computed by means of the MatLab application [21] that computes the inverse of a square matrix using LU factorization.

The AGDC algorithm performs a rigorous analysis and control of the movement vector and of each of its terms, and suitable modifications can be made if any errors are detected, thereby ensuring successful optimization. Once the optimization process has been achieved, the program determines the errors of the optimized parameters [24] and performs an exhaustive analysis of the residuals thus allowing the goodness of fit to be checked [25].

Schematically the AGDC algorithm can be written as follows:

Iteration m \rightarrow Input data: $\mathbf{X}^{(m)}$, SQD $^{(m)}$.

1. Calculate the vector of movement.
 - 1.1. Compute partial numerical derivatives of $(\alpha_j)_C$ with respect to the parameters to be determined \mathbf{X}_p , $(\partial(\alpha_j)_C / \partial X_p)^{(m)}$.
 - 1.2. Compute Gradient vector and Hessian Matrix ($\mathbf{g}^{(m)}$ and $\mathbf{H}^{(m)}$). (33), (34).

- 1.3. Compute $(\mathbf{H}^{(m)})^{-1}$.
- 1.4. Calculate the components of the vector of movement $\mathbf{p}^{(m)}$ (31).
2. Control and correction of the direction of the vector of movement $\mathbf{p}^{(m)}$.
 - 2.1. If $\mathbf{H}^{(m)}$ is singular, $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$, go to 3.
 - 2.2. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} < \varepsilon$ ($\varepsilon =$ scalar close to zero), $\mathbf{p}^{(m)} = -\mathbf{g}^{(m)}$ and go to 3.
 - 2.3. If $\mathbf{p}^{(m)} \mathbf{g}^{(m)} > 0$, $\mathbf{p}^{(m)} = -\mathbf{p}^{(m)}$.
3. Control the length of the vector of movement $\mathbf{p}^{(m)}$.
 - 3.1. Compute the scalar $(\lambda^{(m)})$ by the method of Hartley [3].
 - 3.2. $\mathbf{X}^{(m+1)} = \mathbf{X}^{(m)} + \lambda^{(m)} \mathbf{p}^{(m)}$.
 - 3.3. Determinate the SQD^(m+1) function (29).
 - 3.4. If the Goldstein–Armijo *criterion* [22] is satisfied go to 4.
 - 3.5. $\lambda^{(m)} = \lambda^{(m)}/2$ go to 3.2.
4. Calculate:

$$CON = \left| \frac{SQD^{(m+1)} - SQD^{(m)}}{SQD^{(m)}} \right| \tag{35}$$

5. If convergence is not attained (CON > CC), set $m = m + 1$ and go to 1.
6. $\mathbf{X}^{(m+1)}$ = Optimized Parameters.
7. END optimization.

The procedure followed to carry out the optimization of the activation parameters is schematically expressed as follows:

0. Input data: Experimental data of $(\alpha_j)_E/t_i$, $[B_j]_0$, Convergence Criteria(CC), Initial estimates of the unknown parameters $\mathbf{X}^{(0)}$ ($E_{12}^{(0)}$, $A_{12}^{(0)}$, $\Delta S_{12}^{\neq(0)}$, $\Delta H_{12}^{\neq(0)}$, $E_{23}^{(0)}$, $A_{23}^{(0)}$, $\Delta S_{23}^{\neq(0)}$, $\Delta H_{23}^{\neq(0)}$) (values of *outputs* from ANN application).

I. Optimization of: E_{12} , A_{12} , ΔS_{12}^{\neq} and ΔH_{12}^{\neq} .

Ia. $\mathbf{X} = [E_{12}, A_{12}] \cdot m = 0$. $\mathbf{X}^{(0)} = [E_{12}^{(0)}, A_{12}^{(0)}]$.

Ia1. Calculate $(\alpha_1^{(0)})_C$ (30).

(i) $T = f(t)$ inverse hyperbolic (12) → Mathematical exact solution (14).

(ii) $T = f(t)$ polynomial of n-th degree (15) → Numerical integration of (16) or Numerical solution of the rate differential equation (17).

Ia2. Calculate SQD⁽⁰⁾ (29).

Ia3. **AGDC Algorithm.**

Ia4. Optimized parameters: $\mathbf{X}^* = [E_{12}, A_{12}]$ → Calculate the errors of E_{12} , A_{12} .

Ib. $\mathbf{X} = [\Delta S_{12}^{\neq}, \Delta H_{12}^{\neq}] \cdot m = 0$. $\mathbf{X}^{(0)} = [\Delta S_{12}^{\neq(0)}, \Delta H_{12}^{\neq(0)}]$.

Ib1. Calculate $(\alpha_1^{(0)})_C$ (30): $T = f(t)$ Polynomial of n-th degree (15) → Numerical integration (18) or Numerical solution of the rate differential equation.

Ib2. Calculate SQD⁽⁰⁾ (29).

Ib3. **AGDC Algorithm.**

Ib4. Optimized parameters: $\mathbf{X}^* = [\Delta S_{12}^{\neq}, \Delta H_{12}^{\neq}]$ → Calculate the errors of ΔS_{12}^{\neq} , ΔH_{12}^{\neq} .

II. Optimization of: E_{23} , A_{23} , ΔS_{23}^{\neq} and ΔH_{23}^{\neq} .

$$\text{IIa. } \mathbf{X} = [E_{23}, A_{23}] \cdot m = 0 \cdot \mathbf{X}^{(0)} = \left[E_{23}^{(0)}, A_{23}^{(0)} \right].$$

$$\text{IIa1. Calculate } (\alpha_2(t_i)^{(0)})_C, (\alpha_3(t_i)^{(0)})_C \text{ (30).}$$

$$(i) \text{ Calculate } T_i = \sum_{j=0}^{j=n} a_{j,i} t_i^j$$

$$(ii) \text{ Calculate } k_{12}(T_i) = A_{12} e^{-E_{12}/R} \sum_{j=0}^{j=n} a_j t_i^j$$

$$(iii) \text{ Calculate } k_{23}(T_i) = A_{23}^{(0)} e^{-E_{23}^{(0)}/R} \sum_{j=0}^{j=n} a_j t_i^j$$

(iv) Numerical solution of the rate differential equations (20), (21).

$$\text{IIa2. Calculate SQD}^{(0)} \text{ (29).}$$

IIa3. AGDC Algorithm.

IIa4. Optimized parameters: $\mathbf{X}^* = [E_{23}, A_{23}] \rightarrow$ Calculate the errors of E_{23}, A_{23} .

$$\text{IIb. } \mathbf{X} = [\Delta S_{23}^{\neq}, \Delta H_{23}^{\neq}] m = 0 \cdot \mathbf{X}^{(0)} = [\Delta S_{23}^{\neq(0)}, \Delta H_{23}^{\neq(0)}].$$

$$\text{IIb1. Calculate } (\alpha_2(t_i)^{(0)})_C, (\alpha_3(t_i)^{(0)})_C \text{ (30).}$$

$$(i) \text{ Calculate } T_i = \sum_{j=0}^{j=n} a_{j,i} t_i^j$$

$$(ii) \text{ Calculate } k_{12}(T_i) = (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{12}^{\neq}/R} \sum_{j=0}^{j=n} a_j t_i^j e^{\Delta S_{12}^{\neq}/R}$$

$$(iii) \text{ Calculate } k_{23}(T_i) = (k_B/h) \sum_{j=0}^{j=n} a_j t_i^j e^{-\Delta H_{23}^{\neq(0)}/R} \sum_{j=0}^{j=n} a_j t_i^j e^{\Delta S_{23}^{\neq(0)}/R}$$

(iv) Numerical solution of the rate differential equations (20), (21).

$$\text{IIb2. Calculate SQD}^{(0)} \text{ (29).}$$

IIb3. AGDC Algorithm.

IIb4. Optimized parameters: $\mathbf{X}^* = [\Delta S_{23}^{\neq}, \Delta H_{23}^{\neq}] \rightarrow$ Calculate the errors of $\Delta S_{23}^{\neq}, \Delta H_{23}^{\neq}$.

III. Statistical analysis of Residuals.

3 Computational aspects

The general computational treatment of ANN by means of the application of Matlab “*Neural Networks Toolbox*” with the creation of user’s interfaces (GUI) including the appropriate analysis of Residuals and errors (MSE, SD, etc). In case of the numerical resolutions directly from the beginning of the ordinary differential equations (ODE), we have implemented mainly the followings Matlab functions: (a) ode45, based on an explicit Runge–Kutta formula, the Dormand–Prince pair. It is a one-step solver. In computing $y(t_n)$, it needs only the solution at the immediately preceding time point, $y(t_{n-1})$. (b) ode23s is based on a modified Rosenbrock formula of order 2. Because it is a one-step solver, it can solve some kinds of *stiff* problems for which other are not effective; (c) ode23t is an implementation of the trapezoidal rule using a “free” interpolant (d) ode23tb is an implementation of TR-BDF2, an implicit Runge–Kutta formula with a first stage that is a trapezoidal rule step and a second stage that is a backward differentiation formula of order two and (e) ode15s is a multistep solver, variable order solver based on the numerical differentiation formulas (NDFs). Optionally, it uses the backward differentiation formulas (BDFs, also known as Gear’s method. The numerical integrations have been performed using appropriate *quadrature*

formulas corresponding to a suitable numerical algorithms of resolution of integrals (Simpson, Lobatto, Gauss-Kronrod, Vectorized, etc.). With respect to computational application of AGDC algorithm, the program named KINNOISOT(AGDC), has been designed and performed in our laboratory by means of a computational executable codes (##.m type), in the Matlab environment using “M” language. The program is constituted by a Main program and several Functions or Subroutines.

4 Results and discussion

4.1 Analysis of the functions T/t and α/t

On the basis of the foregoing considerations, different types of functions $T = f(t)$ have been used depending on the differential equation to be solved and/or the group of parameters to be determined. However, all of them have to fulfill the following requirements: (a) the functions must be monotonically increasing, in order to minimize the huge differences existing in the reaction rate of isothermal kinetics between the start and the end of the reaction (b) the increment in temperature produced in the interval of time values of non-isothermal kinetic data acquisition must be the appropriated one to achieve an extent of reaction of 75 % for α values of at least 2 *half-lives*. That leads to choose an appropriate rate of heating for each time value (c) the function must be accessible to experimentation and exactly reproducible in the laboratory. That is, the T values which are provided by the theoretical function $T = f(t)$, have to be reached for each time value of the selected experimental interval.

We have selected the following functions:

1. function of polynomial type of n -th degree (15), without exact mathematical solution for the rate differential equations in which α_2 and α_3 are involved [(24), (25)] and Arrhenius and Eyring equations are considered;
2. inverse function corresponding to a hyperbolic branch (12) only for α_1 and Arrhenius equation, which permits to reach an exact mathematical solution for the rate differential equation.

The profiles of graphic plots of curves synthetically generated from inverse hyperbolic function show that as the value of “ m ” of Eq. (12) decreases a smoothing of the concavity of the hyperbolic branch is observed, tending towards linearity. The suitable value, and the one that satisfies the 3 requisites considered above, is $m = 4.20 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$. We fitted the T/t points to a first-degree polynomial function and then progressively to polynomials from 2nd to 5th degree, observing that the coefficients of the independent variable had negligible values with respect to the value of α_1 from the quadratic term. The values of the Statistical Analysis of Residuals show that the linear fitting can be taken as correct. This means that the T/t data pairs generated with the inverse hyperbolic function with a value of $m = 4.20 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$ can be satisfactorily fitted to a linear function. That is to say, the hyperbolic values of T can be emulated by means of a linear function and can be applied to the case of α_1 values when the Arrhenius equation is considered. At the laboratory, this allowed us to impose a linear heating rate (slope of $0.3734 \text{ K min}^{-1}$), starting ($t = 0$) at $T = 298.15 \text{ K}$.

The profiles of the non-isothermal kinetic curves for all species (α_1/t , α_2/t and α_3/t) sometimes show concave down segments while the isothermal curves are always

concave up. This can be explained in terms of the notion that in non-isothermal curves there are two opposing phenomena that affects the reaction rate: the increase in temperature with time increases the reaction rate, and the logical decrease in the concentration of reagents as they are consumed reduces it. Depending on which of the two phenomena predominates, one concavity or the other will be observed and even when both effects are balanced quasi-linear profiles appear.

4.2 Experimental design (ED)

The application of the computational method ANN demands to perform an appropriate previous organization of the experiments on the basis of the Experimental Design (ED). In the present work, we have chosen the most suitable one, the Central Star Composite Experimental Design (CSCED). The factors are formed by the set of parameters A_{12}/E_{12} and A_{23}/E_{23} for the Arrhenius equation and $\Delta H_{12}^\ddagger/\Delta S_{12}^\ddagger$ and $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ when the Eyring equation is considered. The responses are the non-isothermal kinetic data of the base of the input curves (α_1/t , α_2/t , and α_3/t). It is necessary to consider 2 variables: (a) the extreme values of both factors in each of the 4 sets, which configure the experimental domain and (b) the relative values of each pair of ATP's parameters. Both variables must ensure that the binary combinations of the factors will generate a set of kinetic curves that will have sufficient information to ensure an optimal training process of the neural network. The number of levels of the factors of the ED must be suitable to avoid useless computational work and large differences in the spacing of the values of the responses. Accordingly, to optimize the ANN training process the kinetic curves of the input matrix must have efficient kinetic information and must be correctly distributed according to the choice of a suitable experimental design and an appropriate experimental domain. In agreement with the results of the study of the functions $T = f(t)$, we have generated the non-isothermal kinetic data for both equations (Arrhenius and Eyring) in order to obtain the curves of the input matrix to perform the training process of the neural network from the ATPs organized according to the Experimental Design (ED) (Fig. 1).

4.3 Application of HA: results and discussion

The *training* process of the neural network is crucial in order to know its optimal architecture, which will guarantee the success of the ANN treatment. After testing a **large set** of different network topologies for the treatment of the kinetic system of consecutive reactions, we selected the one that provides the best results since this network architecture is the optimal architecture. The *training* processes, carried out for the 8 parameters by using the computation of the three kinetic curves (α_1/t ; α_2/t and α_3/t), have the following characteristics in common: *input matrices* formed by 45 non-isothermal kinetic curves with 50 pairs of kinetic data (α_1/t) for the first reaction and 100 pairs (α_2/t and α_3/t) for the second one; value of the *random noise* of the order of the experimental error ($\pm 1.0 \cdot 10^{-4}$); time interval to reach a 75 % extent of reaction; function T/t of hyperbolic branch type ($m = 4.2 \cdot 10^{-6} \text{ K}^{-1} \text{ min}^{-1}$), only in the case

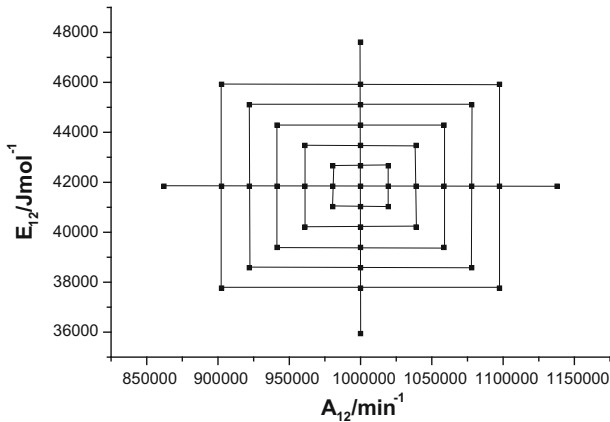


Fig. 1 Experimental Design (CSCED) constituted by 2 factors (A_{12}/E_{12}) and 45 points distributed in 9 levels (points 1–37) and 4 sub-levels: (points 38–45) taking part in the curve base of the *input matrix* used for the *training* process of the neural network

of the first reaction for the Arrhenius equation (A_{12} y E_{12}) and the polynomial type for both reactions and both equations (Arrhenius and Eyring) while the 6 remaining parameters are determined (A_{23} , E_{23} , ΔH_{12}^\ddagger , ΔS_{12}^\ddagger , ΔH_{23}^\ddagger and ΔS_{23}^\ddagger).

We have carried out the *training* of the neuronal networks of the ANN treatment, using the same computational conditions for the 8 parameters grouped in pairs. They are the following ones: regarding to the *back-propagation* algorithm, we selected the Levenberg–Marquardt type for all the parameters because it provides the best optimized results for the *output* and *target* matrices; we have chosen 80/10/10 values for the percentages of computed data on the *training/validation/testing* processes, respectively; we considered a systematic variation of the number of *hidden layers* and their configurations (number of *nodes*) testing a complete set of possibilities for the topology of the neuronal networks in order to strictly determine their optimal architecture.

However we have confirmed the existence of a notorious ambiguity in the attained result that merits a detailed comment to avoid leading to wrong conclusions. Considering the graphic plots of the results of the *training, validation and testing* (“ALL”) processes (Fig. 2) for a neuronal network with optimal structure (100,2,2) and configuration (14–10), we calculated ΔH_{23}^\ddagger and ΔS_{23}^\ddagger . On this graphic, obtained from the simultaneous plot of both parameters, we determined a Regression line: $Output = 1.00 \times Target + 1.30 \cdot 10^{-6}$, with a value of $R = 0.99999$ that corresponds to a good linearity among the elements of the *output/target* matrices. Therefore, these results would be initially acceptable. However, some important considerations must be taken into account: the values of the parameters, which their order of magnitude is very different ($\Delta H_{23}^\ddagger \approx 10^4$ and $\Delta S_{23}^\ddagger \approx 10^2$), are represented on the same scale, so the points show up on the graphic, forming 2 groups separated by a long distance between them. In addition, all the corresponding values to ΔS_{23}^\ddagger are superimposed on the same point. This situation masks the real effects of dispersion, causing that the *output/target* Regression straight and the linear fitting parameters, obtained when both ATP param-

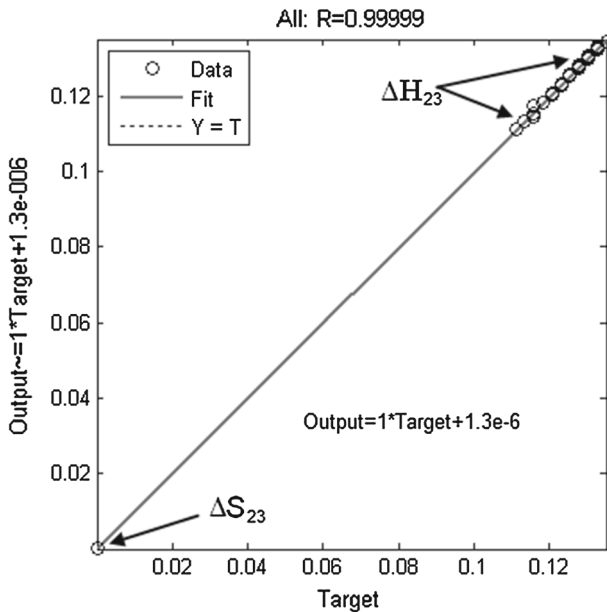


Fig. 2 Plot of the Regression line for the data from the elements of the *output versus target* matrices when both parameters (ΔH_{23}^{\neq} and ΔS_{23}^{\neq}) are represented together for the processes of *training*, *validation* and *testing (ALL)* when ANN is applied to a neural network endowed with an architecture with 2 *hidden* layers (100, 2, 2)

ters are represented together on the graphic, are correct. However, when we consider the *output/target* Regression lines of each separate parameter by using the appropriate scale, a huge dispersion for the case of the ΔS_{23}^{\neq} parameter (Fig. 3, corresponding to the superimposed points observed on the Fig. 2) is detected. The obtained values are: 0.1285 for R^2 , 0.3314 for the slope, (very far away from the value = 1) and 0.0383 for the ordinate at the origin (very different from 0). These values clearly manifest a very poor *output/target* correlation, what makes unacceptable the value of ΔS_{23}^{\neq} determined by ANN. Nevertheless, the results achieved in the ΔH_{23}^{\neq} case prove to have an acceptable linearity as it is shown on the Fig. 4.

In conclusion, these values are unacceptable, so it requires the complete application of HA to improve the results. That is, the obtained values for ATP's parameters after application of ANN method will be the *initial estimates* for the AGDC gradient algorithm, which will provide us very acceptable values for the final ATP's optimized parameters.

The values of the individual Errors and Standard Deviations of each of the 8 ATP's parameter ($SD(b_{ij})$) obtained after application of the ANN, with relative output and target values, can be determined from the following general expression for SD :

$$SD(b_{ij}) = \left(\frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_c} \left[\frac{(b_{ij}^{Outputs} - b_{ij}^{Targets})}{(b_{ij}^{Targets})} \right]^2}{n_p \cdot n_c} \right)^{1/2} \quad (36)$$

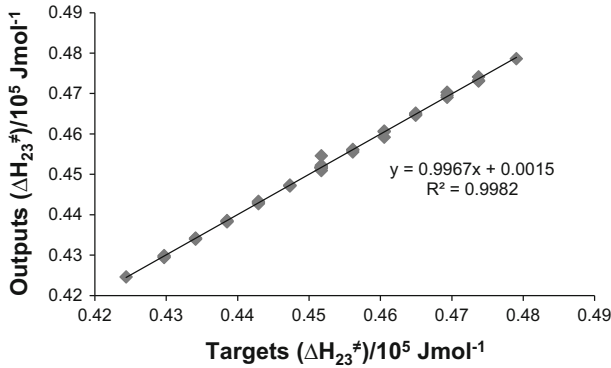


Fig. 3 Plot of the Regression line for the data from the elements of the *output versus target* values when ΔH_{23}^{\pm} is plotted separately for the process of *training, validation and testing (ALL)* of the ANN when the architecture of the neural network is constituted by 2 *hidden layers* (100, 2, 2). The equation obtained is: $Output = 0.9967 * Target + 0.0015 (R^2 = 0.9982)$

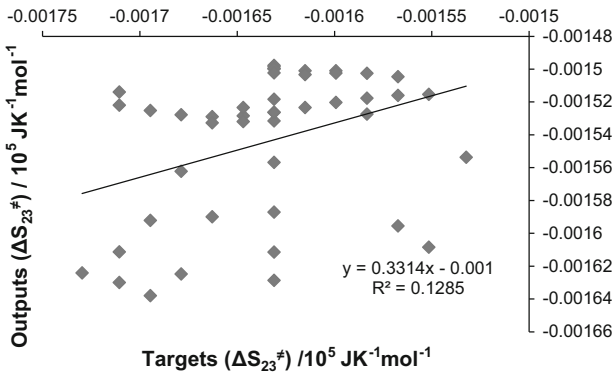


Fig. 4 Plot of the Regression line for the data from the elements of the *output versus target* values when ΔS_{23}^{\pm} is plotted separately for the process of *training, validation and testing (ALL)* of the ANN, when the architecture of the neural network is constituted by 2 *hidden layers* (100, 2, 2). The equation obtained is: $Output = 0.3314 * Target - 0.001 (R^2 = 0.1285)$

where b_{ij} are the ATP’s parameters, n_c is the number of curves and n_p is the number of parameters per curve ($n_p = 1$ for the individual error calculation)

We carried out the *training* process of a large set of neuronal networks considering all the possible configurations for architectures with 2 and 3 *hidden layers*, discarding those of a single layer. In light of the high number of tested cases, we just only show the best results on the Tables 1 and 2. All of them were obtained from the most significant configurations of the curves corresponding to the *central point* of the 45 curves of the ED. The value pairs of *outputs*, A_{23}/E_{23} and $\Delta H_{23}^{\pm}/\Delta S_{23}^{\pm}$, and the deviations in % (Dev.%) respect to the values used to generate all the data, are showed on the first part of the Tables 1 and 2. In addition, the second part of the Tables shows the values of the optimized parameters after applying AGDC and the deviations in % (Dev.%) of the real values. After completing the whole application of HA we observed a great

Table 1 Results of the *training* process of neural networks constituted by architectures with 2 and 3 *hidden* layers and several configurations, when E_{23} and A_{23} are computed

Config	ANN				ANN + AGDC			
	A_{23} (10^5 min^{-1})	E_{23} (10^4 J mol^{-1})	Dev.% (A_{23})	Dev.% (E_{23})	A_{23} (10^5 min^{-1})	E_{23} (10^4 J mol^{-1})	Dev.% (A_{23})	Dev.% (E_{23})
10/10	4.9626	4.2496	0.7466	0.0071	4.9740	4.2486	0.5207	0.0318
10/11	5.0512	4.2622	-1.0254	-0.2891	5.1192	4.2561	-2.3840	-0.1439
10/12	4.9752	4.3018	0.4959	-1.2200	5.0831	4.2543	-1.6619	-0.1003
10/14	4.9690	4.2858	0.6190	-0.8433	5.1641	4.2584	-3.2818	-0.1966
13/10	5.0672	4.2647	-1.3457	-0.3472	5.1469	4.2575	-2.9384	-0.1769
15/10	4.9809	4.2945	0.3814	-1.0470	5.1079	4.2356	-2.1979	-0.1326
20/10	5.0490	4.3250	-0.9813	-1.7665	5.0012	4.2501	-0.0235	-0.0016
7/10/10	5.0028	4.3079	-0.0576	-1.3625	5.0838	4.2543	-1.6762	-0.1017
9/10/10	5.1849	4.1884	-3.6988	1.4479	4.6510	4.2313	6.9801	0.4399

This table shows the obtained results of the ATP's parameters and their % deviations (Dev.%) when applying the algorithms ANN and HA (ANN+AGDC)

improvement just by comparing the values of Dev.%, (columns 8th and 9th) of both parameters with the ones calculated by the ANN application (columns 4th and 5th).

The improvement in the values of ATP's parameters is much better appreciated on a joint plot (Fig. 5) of the Residuals of α_2 , calculated by using the values of parameters $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$, which were obtained from the ANN application (37), and the values determined by the HA algorithm application (38). In Fig. 5 we can appreciate this improvement, which maximum value is produced when a 40% extent of reaction is reached by comparing the values of $Res(\alpha)_{HA} = 0.0013$ and $Res(\alpha)_{ANN} = 0.388$.

$$[Res(\alpha)_{ANN} = \alpha_{synt} - \alpha_{cal}(ANN)] \quad (37)$$

$$[Res(\alpha)_{HA} = \alpha_{synt} - \alpha_{cal}(HA)] \quad (38)$$

The *prediction* process, necessary to determine the final optimized parameters, consists in the application of HA to new kinetic curves, different from the considered ones at the *training* processes of neuronal networks. These new ones have been generated from ATP value pairs within the maximum intervals defined in the ED. The *prediction* constitutes the final process of the HA application to finally determine the unknown ATP parameters, corresponding to experimental kinetics carried out at the Laboratory. The obtained results are the value pairs of the parameters E_{23}/A_{23} (Table 3) and $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ (Table 4). As for the *prediction* process, carried out after the ANN application, we used the optimal neuronal network after carrying out exhaustive processes of *training*, computing a base of 8 kinetic curves, the most suitable number to perform a multiple and simultaneous *prediction* of the ANN initial stage. After the AGDC algorithm application, we determine the final optimized values of the parameters, resulting from the complete application of the HA algorithm and the Dev.% between the Optimized Values and the ones used in the kinetic generation of data. After analyzing all

Table 2 Results of the *training* process for neural networks constituted by architectures with 2 and 3 *hidden* layers and several configurations, when ΔH_{23}^\ddagger and ΔS_{23}^\ddagger are computed

Config	ANN				ANN+AGDC			
	ΔS_{23}^\ddagger (J K ⁻¹ mol ⁻¹)	ΔH_{23}^\ddagger (10 ⁴ J mol ⁻¹)	Dev.% (ΔS_{23}^\ddagger)	Dev.% (ΔH_{23}^\ddagger)	ΔS_{23}^\ddagger (J K ⁻¹ mol ⁻¹)	ΔH_{23}^\ddagger (10 ⁴ J mol ⁻¹)	Dev.% (ΔS_{23}^\ddagger)	Dev.% (ΔH_{23}^\ddagger)
8/10	-163.7169	4.5016	-0.3782	0.3451	-163.6531	4.5000	-0.3391	0.3811
10/9	-164.0223	4.4914	-0.5655	0.5704	-163.9694	4.4902	-0.5334	0.5972
10/11	-160.5622	4.4979	1.5559	0.4251	-162.2034	4.5449	0.5498	-0.6142
10/12	-179.0484	4.4927	-9.7783	0.5410	-164.6662	4.4685	-0.9603	1.0782
10/15	-136.3193	4.5143	16.4197	0.0625	-158.9482	4.6457	2.5455	-2.8456
10/20	-178.3090	4.5212	-9.3250	-0.0898	-162.5956	4.5329	0.3093	-0.3482
13/10	-160.4780	4.5002	1.6075	0.3752	-162.1255	4.5774	0.5975	-0.6679
14/10	-163.7297	4.5430	-0.3861	-0.5714	-163.0215	4.5196	0.0481	-0.0531
15/10	-161.7405	4.5239	0.8335	-0.1499	-162.3150	4.5416	0.4813	-0.5405
9/10/10	-161.9950	4.5061	0.6774	0.4363	-162.7351	4.5285	0.2237	-0.2508
10/9/10	-181.5295	4.5271	-11.2995	0.5704	-159.9146	4.6161	1.9530	-2.1899
10/10/10	-162.6242	4.5210	0.2916	0.1582	-162.7918	4.5268	-0.1889	-0.2125
10/10/13	-161.8261	4.4894	0.7810	-0.0898	-162.9366	4.5222	0.1002	-0.1109
10/10/14	-160.3689	4.5145	1.6744	-0.7670	-161.8253	4.5567	0.7815	-0.8754
10/12/10	-163.6233	4.5064	-0.3208	-0.1499	-163.5313	4.5038	-0.2644	0.2973
10/15/10	-165.0704	4.5121	-1.2081	0.1113	-164.1942	4.4830	-0.6709	0.7566
10/20/10	-182.9596	4.5263	-12.1763	-0.2026	-161.2712	4.5745	1.1213	-1.2682

This table shows the obtained results of the ATP's parameters and their % deviations (Dev.%) when applying the algorithms ANN and HA (ANN+AGDC)

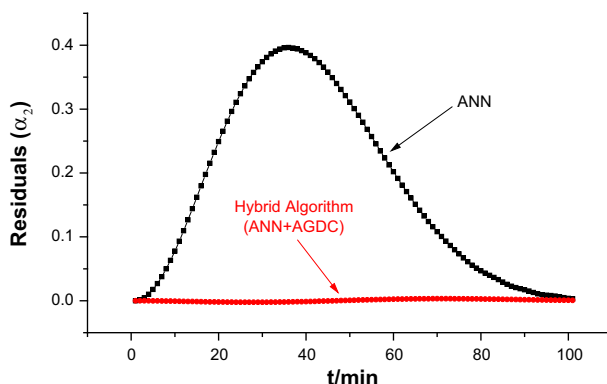


Fig. 5 Plot of the *Residuals* of α_2 using the values of the ATP's parameters (ΔH_{23}^\ddagger and ΔS_{23}^\ddagger) obtained by applying ANN method [Eq. (37)] and those obtained from Eq. (38) after the application of the complete algorithm HA

Table 3 Results of the 4 *prediction* processes in the case of the optimal architecture of neural network (100, 2, 2) when Arrhenius equation is considered and A_{23}/E_{23} values are predicted, after application of HA (ANN+AGDC)

HA Algorithm (ANN + AGDC)					
Real values (A_{23}) (10^5 min^{-1})	Real values (E_{23}) (10^4 J mol^{-1})	Optimized values (A_{23}) (10^5 min^{-1})	Optimized values (E_{23}) (10^4 J mol^{-1})	Dev.% (A_{23})	Dev.% (E_{23})
5.0830	4.0306	5.7654	5.3756	30.0620	-17.3260
4.9202	4.1031	3.6072	4.1655	27.8558	1.9876
5.0173	4.0928	5.0621	4.0925	-3.2565	5.5473
4.9789	4.2181	5.0090	3.9874	5.3773	0.3499

Real and optimized values can be compared after evaluation of the values of their deviations (Dev.%)

Table 4 Results of the 8 *prediction* processes in the case of the optimal architecture of neural network (100, 2, 2) when Eyring equation is considered and $\Delta H_{23}^\ddagger/\Delta S_{23}^\ddagger$ values are predicted, after application of HA (ANN+AGDC)

HA algorithm (ANN+ AGDC)					
Real values (ΔS_{23}^\ddagger) ($\text{J K}^{-1} \text{ mol}^{-1}$)	Real values (ΔH_{23}^\ddagger) (10^4 J mol^{-1})	Optimized values (ΔS_{23}^\ddagger) ($\text{J K}^{-1} \text{ mol}^{-1}$)	Optimized values (ΔH_{23}^\ddagger) (10^4 J mol^{-1})	Dev.% (ΔS_{23}^\ddagger)	Dev.% (ΔH_{23}^\ddagger)
-159.5642	4.3408	-158.1711	4.2976	-0.9136	0.9999
-163.8140	4.7031	-166.6814	4.7811	1.6402	-1.8692
-152.7474	4.3832	-155.5510	4.4701	1.7222	-1.9387
-150.1146	4.4330	-156.0480	4.5482	2.4209	-2.6887
-159.0111	4.6072	-163.3297	4.6994	1.7745	-2.0446
-150.1152	4.4734	-158.5474	4.5650	1.8465	-2.0531
-153.2067	4.5551	-159.5911	4.7207	3.0962	-3.4960
-152.6486	4.5172	-1579356	4.6775	3.1664	-3.5481

Real and optimized values can be compared after evaluation of the values of their deviations (Dev.%)

Table 5 Final values of the 8 ATP's parameters

(A_{12} , E_{12} , A_{23} , E_{23} , ΔH_{12}^\ddagger , ΔS_{12}^\ddagger , ΔH_{23}^\ddagger and ΔS_{23}^\ddagger) obtained after application of the HA and their optimal architecture and configurations (*nodes*) of the neural networks used in the ANN treatment

(1) min^{-1} , (2) J mol^{-1} , (3) J mol^{-1} , (4) $\text{J mol}^{-1}\text{K}^{-1}$

ATP	Optimum architecture	Nodes	HA final values
A_{12}	(50,3,2)	10/20/10	1.10^6 (1)
E_{12}	(50,3,2)	10/20/10	41840 (2)
ΔH_{12}^\ddagger	(50,3,2)	10/9/10	38285 (3)
ΔS_{12}^\ddagger	(50,3,2)	10/9/10	-172.7 (4)
A_{23}	(100,2,2)	20/10	5.10^5 (1)
E_{23}	(100,2,2)	20/10	42500 (2)
ΔH_{23}^\ddagger	(100,2,2)	14/10	45172 (3)
ΔS_{23}^\ddagger	(100,2,2)	14/10	-163.1 (4)

the values of the *Statistic moments* of errors [$SD(b_{ij})$] and Dev.%, according to Eq. (37), and (38), we can easily know the optimal architecture in terms of structure and configuration of the *hidden* layers used to determine the 8 ATP parameters and their values (Table 5).

5 Conclusions

The obtained results of the HA application demonstrate its great robustness when it is applied to the calculation of the ATP's parameters of the kinetic system of the first order consecutive reactions. Additionally, the HA embodies a satisfying versatility, providing evidences of its applicability to other models and kinetic systems in order to obtain other kind of kinetic parameters. The great advantage of HA is the efficient resolution of the ambiguity of the results determined by ANN: not only discriminates between the pairs of parameters obtained in the *output* matrices, but also calculates their individual values with considerable accuracy and precision. Furthermore, the exhaustive control and corrections performed over the movement vector, solves other weak point of the classic optimization algorithms avoiding the possibility of being trapped at local minima, saddle points...etc. This allows to AH successfully determine the values of the 8 ATP's parameters involved in the studied kinetic system (A_{12} , E_{12} , A_{23} , E_{23} , ΔH_{12}^\ddagger , ΔS_{12}^\ddagger , ΔH_{23}^\ddagger and ΔS_{23}^\ddagger), of which high-number of parameters and very different orders of magnitude make the task extremely complicated for the classic algorithms of mathematical optimization.

The lack of necessity to know a priori the order of magnitude and the sign of the 8 parameters represents another important advantage. Therefore, the initial ANN treatment provides the values of the all ATP's parameters used by AGDC as *initial estimates*. In addition, the computational application, in which the HA is implemented, doesn't present difficulties to non-expert users.

Finally, in comparison with the isothermal kinetics, the evaluation of non-isothermal kinetics produces evident advantages in the experimental aspect, generating significant savings both in reagents and laboratory time. This allows one to perform only one replicated non-isothermal kinetic since it is sufficient for the computational treatment with the HA proposed in this paper.

References

1. A.A. Frost, R.G. Pearson, *Kinetics and Mechanism* (Wiley, New York, 1961)
2. C.H. Bandford, C.F.H. Tipper, *Comprehensive Chemical Kinetics* (Elsevier, Amsterdam, 1973)
3. M.A. Wolfe, *Numerical Methods for Unconstrained Optimization* (Van Nostrand, New York, 1978)
4. S. Vajda, H. Rabitz, *J. Phys. Chem.* **92**, 701–707 (1988)
5. S. Vajda, H. Rabitz, *J. Phys. Chem.* **98**, 5265–5271 (1994)
6. J. Zupan, J. Gasteiger, *Neural Networks for Chemists* (VCH, Weinheim, 1993)
7. M.M. Canedo, J.L. González-Hernández, *Chemom. Intell. Lab. Syst.* **66**, 63–78 (2003)
8. M.M. Canedo, J.L. González-Hernández, *Chemom. Intell. Lab. Syst.* **66**, 93–97 (2003)
9. M.M. Canedo, J.L. González-Hernández, *J. Math. Chem.* **49**, 163–184 (2011)
10. M.M. Canedo, J.L. González-Hernández, S. Encinar, *Appl. Math. Comput.* **219**, 7089–7101 (2013)
11. J.L. González-Hernández, M.M. Canedo, C. Grande Martín, *MATCH Commun. Comput. Chem.* **70**, 951–970 (2013)
12. A. Niazi, R. Leardi, *J. Chemom.* **26**, 345–351 (2012)
13. M. Maeder, Y.M. Neuhold, G. Puxty, *Chemom. Intell. Lab. Syst.* **70**, 193–203 (2004)
14. C. Hervás, J.A. Algar, M. Silva, *J. Chem. Inf. Comput. Sci.* **40**, 724–731 (2000)
15. S. Curteanu, H. Cartwright, *J. Chemom.* **25**, 527–549 (2011)
16. F. Amato, J.L. González-Hernández, J. Havel, *Talanta* **93**, 72–78 (2012)
17. S. Encinar, J.L. González-Hernández, M. Mar Canedo, *J. Math. Chem.* **51**, 1634–1653 (2013)
18. S. Encinar, J.L. González-Hernández, M. Mar Canedo, *MATCH Commun. Comput. Chem.* **72**, 427–450 (2014)
19. K.J. Laidler, *Pure Appl. Chem.* **68**, 149–192 (1996)
20. T. Kohonen, *Neural Networks.* **1**, 3–16 (1988)
21. MathWorks MatLab, R2012a, Vs7.14.0.739 (2012)
22. P. Gill, W. Murray, M.H. Wright, *Practical Optimization* (Academic Press, London, 1981)
23. C.F. Gerald, P.O. Wheatley, *Applied Numerical Analysis* (Adinson-Wesley, Massachusetts, 1984)
24. J. Topping, *Errors of Observation and Their Treatment* (Chapman and Hall, London, 1978)
25. R.D. Cook, S. Weisberg, *Residuals and Influence in Regression* (Chapman and Hall, New York, 1982)