

ANEXO IV: PROGRAMAS MATEMÁTICOS UTILIZADOS

1- MÉTODOS NUMÉRICOS PARA BÚSQUEDA DE RAÍCES.

1.1- MÉTODO DE LA BISECCIÓN

```
f[x_] = ;
x0 = ;
y0 = ;
k = ;
e0 = ;
z0 = (x0 + y0) / 2;

If[Sign[f[x0]] ≠ Sign[f[y0]],
  For[i = 0, i < k, i++,
    If[Sign[f[xi]] ≠ Sign[f[zi]],
      xi+1 = xi;
      yi+1 = zi;
      zi+1 = (xi+1 + yi+1) / 2,
      xi+1 = zi;
      yi+1 = yi;
      zi+1 = (xi+1 + yi+1) / 2]];
Do[e1 = Abs[(zi+1 - zi) / zi+1] * 100, {i, 0, k}];
Table[{xi+1, N[zi, 10], N[e1, 10]}, {i, 0, k}] // TableForm, Print
["El método de la bisección no es aplicable
porque f(x0) y f(y0) tienen el mismo signo"]]
```

1.2- MÉTODO DE LA REGLA FALSA

```

f[x_] :=;
x0 =;
y0 =;
k =;
e0 =;

z0 = y0 - f[y0] * (x0 - y0) / (f[x0] - f[y0]);

If[Sign[f[x0]] ≠ Sign[f[y0]],
  For[i = 0, i < k, i++,
    If[Sign[f[xi]] ≠ Sign[f[zi]],
      xi+1 = xi;
      yi+1 = zi;
      zi+1 = yi+1 - f[yi+1] * (xi+1 - yi+1) / (f[xi+1] - f[yi+1]),
      xi+1 = zi;
      yi+1 = yi;
      zi+1 = yi+1 - f[yi+1] * (xi+1 - yi+1) / (f[xi+1] - f[yi+1])];
  Do[e1+1 = Abs[(zi+1 - zi) / zi+1] * 100, {i, 0, k}];
  Table[{xr1+1, N[zi, 10], N[e1, 10]}, {i, 0, k}] // TableForm, Print
  ["El método de la regla falsa no es aplicable
   porque f(x0) y f(y0) tienen el mismo signo"]]
```

1.3- MÉTODO DE LA SECANTE

```

f[x_] :=;
x0 =;
x1 =;
k =;
e0 =;
Do[xn+1 = xn - f[xn] * (xn - xn-1) / (f[xn] - f[xn-1]), {n, 1, k}];
Do[e1+1 = Abs[(xi+1 - xi) / xi+1] * 100, {i, 0, k}];
Table[{i, N[xi, 9], N[f[xi], 9], N[e1, 10]}, {i, 0, k}] // TableForm
```

2- POLINOMIO INTERPOLADOR DE NEWTON.

```

X0 = 1
X1 = 2
X2 = 3
X3 = 4
X4 = 5
X5 = 6
X6 = 7
X7 = 8
X8 = 9
X9 = 10
X10 = 11
X11 = 12

Y0 = 1
Y1 = 2
Y2 = 3
Y3 = 4
Y4 = 5
Y5 = 6
Y6 = 7
Y7 = 8
Y8 = 9
Y9 = 10
Y10 = 11
Y11 = 12

n = 11

Do [ Bi = (Yi - Yi-1) / (Xi - Xi-1), {i, 1, n} ];
Do [ Ci = (Bi+1 - Bi) / (Xi+1 - Xi-1), {i, 1, n - 1} ];
Do [ Di = (Ci+1 - Ci) / (Xi+2 - Xi-1), {i, 1, n - 2} ];
Do [ Ei = (Di+1 - Di) / (Xi+3 - Xi-1), {i, 1, n - 3} ];
Do [ Fi = (Ei+1 - Ei) / (Xi+4 - Xi-1), {i, 1, n - 4} ];
Do [ Gi = (Fi+1 - Fi) / (Xi+5 - Xi-1), {i, 1, n - 5} ];
Do [ Hi = (Gi+1 - Gi) / (Xi+6 - Xi-1), {i, 1, n - 6} ];
Do [ Ii = (Hi+1 - Hi) / (Xi+7 - Xi-1), {i, 1, n - 7} ];
Do [ Ji = (Ii+1 - Ii) / (Xi+8 - Xi-1), {i, 1, n - 8} ];
Do [ Ki = (Ji+1 - Ji) / (Xi+9 - Xi-1), {i, 1, n - 9} ];
Do [ Li = (Ki+1 - Ki) / (Xi+10 - Xi-1), {i, 1, n - 10} ];

```

```

Do[Bi = (yi - yi-1) / (xi - xi-1), {i, 1, n}];
Do[Ci = (Bi+1 - Bi) / (xi+1 - xi-1), {i, 1, n-1}];
Do[Di = (Ci+1 - Ci) / (xi+2 - xi-1), {i, 1, n-2}];
Do[Ei = (Di+1 - Di) / (xi+3 - xi-1), {i, 1, n-3}];
Do[Fi = (Ei+1 - Ei) / (xi+4 - xi-1), {i, 1, n-4}];
Do[Gi = (Fi+1 - Fi) / (xi+5 - xi-1), {i, 1, n-5}];
Do[Hi = (Gi+1 - Gi) / (xi+6 - xi-1), {i, 1, n-6}];
Do[Ii = (Hi+1 - Hi) / (xi+7 - xi-1), {i, 1, n-7}];
Do[Ji = (Ii+1 - Ii) / (xi+8 - xi-1), {i, 1, n-8}];
Do[Ki = (Ji+1 - Ji) / (xi+9 - xi-1), {i, 1, n-9}];
Do[Li = (Ki+1 - Ki) / (xi+10 - xi-1), {i, 1, n-10}];

f[X_] = y0 + B1 * (X - x0) + C1 * (X - x0) * (X - x1)
+ D1 * (X - x0) * (X - x1) * (X - x2)
+ E1 * (X - x0) * (X - x1) * (X - x2) * (X - x3)
+ F1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4)
+ G1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4) * (X - x5)
+ H1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4) * (X - x5) * (X - x6)
+ I1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4) * (X - x5) * (X - x6) * (X - x7)
+ J1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4) * (X - x5) * (X - x6) * (X - x7) * (X - x8)
+ K1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4) * (X - x5) * (X - x6) * (X - x7) * (X - x8) * (X - x9)
+ L1 * (X - x0) * (X - x1) * (X - x2) * (X - x3) * (X - x4) * (X - x5) * (X - x6) * (X - x7) * (X - x8) * (X - x9) * (X - x10)

Print["El polinomio de Newton para los datos es:"]
f[X]
Print["La gráfica de los puntos y el polinomio de interpolación de Newton se representan:"]
datos := {{x0, y0}, {x1, y1}, {x2, y2}, {x3, y3}, {x4, y4}, {x5, y5}, {x6, y6}, {x7, y7}, {x8, y8}, {x9, y9},
{x10, y10}, {x11, y11}};
puntos :=
ListPlot[datos, PlotStyle -> PointSize[0.02], DisplayFunction -> Identity];
curva := Plot[f[X], {X, Min[x0, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11],
Max[x0, x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11], DisplayFunction -> Identity];
Show[puntos, curva, DisplayFunction -> $DisplayFunction];

```