Results in Applied Mathematics 21 (2024) 100417

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics

A variable stepsize hybrid block optimized technique for integrating a class of singularly perturbed parabolic problems

Mufutau Ajani Rufai [a,*], Higinio Ramos [b,c], Bruno Carpentieri [a,*]

[a] Faculty of Engineering, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
[b] Scientific Computing Group, Universidad de Salamanca, Plaza de la Merced 37008 Salamanca, Spain
[c] Escuela Politécnica Superior de Zamora, Campus Viriato, 49022 Zamora, Spain

ARTICLE INFO

MSC:
65L05
65L20

Keywords:
Hybrid block technique
Singularly perturbed problem
Parabolic convection–diffusion problem
Adaptive formulation

ABSTRACT

This paper presents and successfully applies an optimized hybrid block technique using a variable stepsize implementation to integrate a type of singularly perturbed parabolic convection–diffusion problems. The problem under consideration is semi-discretized by utilizing the method of lines. A few numerical experiments have been presented to ascertain the proposed error estimation and adaptive stepsize strategy. Furthermore, the comparison of the proposed method with other techniques in the literature is conducted via numerical experiments, and the results show that our method outperforms other existing methods.

1. Introduction

Consider a singularly perturbed parabolic convection–diffusion problem (SPPCDP) that is expressed as:

$$y_t - L_{x,\epsilon}(t)y = f(x,t), \quad (x,t) \in D = [0,1] \times [0,T], \tag{1}$$

with

$$L_{x,\epsilon}(t)y \equiv \epsilon y_{xx} + b(x,t)y_x + d(x,t)y.$$

The above problem is subject to the initial and boundary conditions:

$$y(x,0) = V(x) \text{ on } Q_x = [0,1] \times \{0\}, \tag{2}$$

$$y(0,t) = q_0(t) \text{ on } Q_0 = \{0\} \times [0,T], \tag{3}$$

$$y(1,t) = q_1(t) \text{ on } Q_1 = \{1\} \times [0,T], \tag{4}$$

where the boundary layer is located near the boundary Q_x, and $0 < \epsilon \ll 1$ is a perturbation parameter, Q_0, Q_1, and Q_x correspond to the left, right, and bottom boundaries of the domain D.

We note that the SPPCDP in (1) is a problem with multi-scale solutions, in which some parts of the solution vary smoothly while others vary rapidly. It arises in problems involving convection and diffusion phenomena, where the governing equation involves a

* Corresponding authors.
E-mail addresses: mufutauajani.rufai@unibz.it (M.A. Rufai), higra@usal.es (H. Ramos), bruno.carpentieri@unibz.it (B. Carpentieri).

https://doi.org/10.1016/j.rinam.2023.100417
Received 23 September 2023; Received in revised form 6 November 2023; Accepted 24 November 2023

combination of diffusion, convection, and reaction terms. The parameter (ϵ) in Eq. (1) represents the perturbation parameter that determines the relative importance of the diffusion term compared to the convection term.

The singularly perturbed partial differential problem represented by Eqs. (1)–(4) finds wide-ranging applications across diverse fields of engineering and applied sciences. This problem plays a crucial role in simulating real-life phenomena in areas such as aerodynamics, lubrication theory, heat transfer, biochemical reactions, physics, fluid dynamics, and engineering. Moreover, the formulation presented in Eq. (1) captures various practical modeling applications, including semiconductor device modeling, turbulence models, laminar flow thermal boundary layer models, and heat flow modeling. Further insights, including the mentioned applications of SPPCDP in various science and engineering fields, can be found in [1–7] and references therein. These sources provide valuable insights into the practical significance of the SPPCDP and its relevance for modeling numerous real-life problems in science-related areas and engineering domains.

As consequence of the presence of the small parameter (ϵ) in (1), the solution of (1)–(4) exhibits a boundary layer behavior, which is characterized by rapid changes in a narrow region near the domain boundary. The thickness of this layer is proportional to $\sqrt{\epsilon}$, which is small when ϵ is small. The convection term $b(x,t)y_x$ in this problem can cause an advection of the solution towards or away from the boundaries, while the diffusion term ϵy_{xx} can originate sharp gradients in the solution. The coefficient $d(x,t)$ denotes a source or sink term that can affect the overall behavior of the solution. The initial and boundary conditions specify the values of $y(x,t)$ at different parts of the boundary domain. The initial condition $y(x,0) = V(x)$ describes how the solution is initially distributed throughout the domain, while the boundary conditions $y(0,t) = q_0(t)$ and $y(1,t) = q_1(t)$ describe how the solution enters or exits through the boundaries. These features can lead to a complex behavior in the solution, such as oscillations and sharp gradients near the boundaries. Therefore, analytical or numerical approaches for integrating the SPPCDP in (1) and related problems must be carefully developed to capture these features and maintain and preserve stability and efficiency. For more detailed information on the theory and features of singularly perturbed differential problems, we refer the reader to [5–7].

Numerous occasions necessitate applying numerical techniques for solving SPPCDP, either because an analytical solution is unknown or has no reasonable or practical meaning. Numerical strategies give a powerful alternative for solving SPPCDP.

To cite some of the available numerical methods for giving solutions to the SPPCDP in (1) and similar problems, we can cite a quintic B-spline method in [8], the parameter-robust numerical method in [9], the piecewise-uniform mesh approach by Roos et al. [6], the finite element method presented by Constantinou and Xenophontos [10], the layer-adapted strategy reported by Kumar [11], the hybrid block method proposed by Rufai [12], the parameter uniform difference scheme by Selvi and Ramanujam [13], the difference approximations in [14], a high order convergent numerical method in [15], the collocation method in [16], the continuous block method reported by Duromola et al. [17], a numerical simulation technique by Arora and Joshi [18], a parametric method in [19], the uniform and accelerated parameter numerical methods in [20,21], optimized Nyström methods in [22,23], the finite difference method in [24], or the numerical methods reported in [25–28].

We should point out that most of the approaches for solving (1) cited above used constant step-size implementations (CSSI). Since the solution to the problem under consideration exhibits a multiscale behavior in the integration interval, most numerical methods with CSSI are expected to perform poorly, particularly when considering a large integration interval. In this manuscript, we describe a variable step size implementation utilizing an optimized hybrid block technique (OHBT) to overcome this issue and efficiently solve problem (1).

This manuscript is organized as follows: In Section 2, the SPPCDP is converted into a system of ODEs using the method of lines. The proposed method is presented in Section 3, the theoretical analysis is provided in Section 4, an error estimation of the proposed method is explained in Section 5. In Section 6, an implementation of the proposed method is described, in Section 7, numerical experiments are discussed, and a brief conclusion is given in the last section.

2. Problem transformation

In this section, the singularly perturbed parabolic convection-diffusion problem, as presented in Eq. (1), will be transformed into a set of ordinary differential equations (ODEs) using the method of lines for approximating the spatial derivatives by suitable finite difference schemes. We begin by dividing the spatial domain into a grid comprising N evenly spaced subintervals. This grid is given by $x_i = i\,\Delta x$, where $i = 0, 1, \ldots, N$, and the spacing between adjacent points is represented by $\Delta x = 1/N$.

Next, we approximate the second-order derivative in space using a finite difference scheme. A common choice is the centered difference approximation:

$$y_{xx}(x_i,t) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2},$$

where $y_i = y(x_i,t)$.

Similarly, we approximate the first-order derivative in space using the centered finite difference scheme:

$$y_x(x_i,t) \approx \frac{y_{i+1} - y_{i-1}}{2\Delta x}.$$

Substituting these approximations into the original PDE yields:

$$\frac{dy_i}{dt} = \epsilon\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} + b(x_i,t)\frac{y_{i+1} - y_{i-1}}{2\Delta x} + d(x_i,t)y_i + f(x_i,t).$$

We see that through the method of lines, for each interior point at the spatial grid, we obtain the following system of ODEs for $i = 1, 2, \ldots, N-1$:

$$\begin{aligned}
\frac{dy_1}{dt} &= \frac{\epsilon(y_2 - 2y_1 + y_0)}{\Delta x^2} + \frac{b(x_1,t)(y_2 - y_0)}{2\Delta x} + d(x_1,t)y_1 + f(x_1,t),\\
&\vdots\\
\frac{dy_{N-1}}{dt} &= \frac{\epsilon(y_N - 2y_{N-1} + y_{N-2})}{\Delta x^2} + \frac{b(x_{N-1},t)(y_N - y_{N-2})}{2\Delta x} + d(x_{N-1},t)y_{N-1} + f(x_{N-1},t),
\end{aligned}$$

where the values y_0 and y_N are specified by the boundary conditions.

The vector form of the above system can be expressed as follows:

$$\frac{d\mathbf{y}}{dt} = \mathbf{A}\mathbf{y} + \mathbf{f}(t), \tag{5}$$

where $\mathbf{y} = [y_1, \ldots, y_{N-1}]^T$ is a vector of the numerical approximations to $y(x_i, t)$ at time t, $\mathbf{A}$ is an $(N-1)\times(N-1)$ matrix given by

$$\mathbf{A} = \frac{\epsilon}{\Delta x^2}\,\mathbf{C} + \frac{1}{2\Delta x}\,\mathbf{B} + \mathbf{D},$$

where $\mathbf{C}, \mathbf{B}$ are $(N-1)\times(N-1)$ tridiagonal matrices given by

$$C = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 0\\ 1 & -2 & 1 & \cdots & 0 & 0\\ 0 & 1 & -2 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & -2 & 1\\ 0 & 0 & 0 & \cdots & 1 & -2 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & b(x_1,t) & 0 & \cdots & 0 & 0\\ -b(x_2,t) & 0 & b(x_2,t) & \cdots & 0 & 0\\ 0 & -b(x_3,t) & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 0 & b(x_{N-2},t)\\ 0 & 0 & 0 & \cdots & -b(x_{N-1},t) & 0 \end{bmatrix},$$

and $\mathbf{D}$ is a diagonal matrix with entries given by the coefficients $d(x_i, t), i = 1, 2, \ldots, N-1$.

Finally,

$$\mathbf{f}(t) = \left[\left(\frac{\epsilon}{\Delta x^2} - \frac{b(x_1,t)}{2\Delta x}\right) y_0 + f(x_1,t), f(x_2,t), \ldots, f(x_{N-2},t), \left(\frac{\epsilon}{\Delta x^2} + \frac{b(x_{N-1},t)}{2\Delta x}\right) y_N + f(x_{N-1},t)\right]^T$$

is a vector related to the boundary conditions and the evaluation of f at selected spatial grid points. Furthermore, the initial values at $t = 0$ can be incorporated into the system of ODEs by specifying appropriate values for the entries of $\mathbf{y}$ at time $t = 0$ at the grid points of the spatial domain. The resulting system of ODEs can then be solved numerically using an ODE solver to obtain the numerical solution $\mathbf{y}(t)$ at each time step. The solution $\mathbf{y}(t)$ can then be used to approximate the solution $y(x_i, t)$ of the original PDE at time t. For solving the resulting ODE system we will consider the block method derived in the following section.

3. Derivation of the proposed block method

Following the optimization procedure in [29,30], we begin the derivation of the block method by assuming that the solution of a first order differential equation of the form $y'(t) = f(t, y(t))$ can be approximated by the following polynomial $\rho(t)$, with $t_{n+1} = t_n + h$ and h is the stepsize

$$y(t) \simeq \rho(t) = \sum_{k=0}^{4} a_k\, t^k, \tag{6}$$

whose first derivative is

$$y'(t) \simeq \rho'(t) = \sum_{k=1}^{4} a_k k t^{k-1}, \tag{7}$$

where the coefficients a_k represent unknown values that must be determined when collocation conditions are imposed at suitable points.

The OHBT method is developed based on the optimization of two points, namely $t_{n+c_1} = t_n + c_1 h$ and $t_{n+c_2} = t_n + c_2 h$, within the interval $[t_n, t_{n+1}]$. We proceed to evaluate the equation in (6) at t_n and the equation in (7) at t_n, t_{n+c_1}, t_{n+c_2}, and t_{n+1}. This leads to a system of five equations comprising the five unknowns, a_k for $k = 0, 1, \ldots, 4$. The equations are as follows: The first equation sets the value of the function ρ at the point t_n equal to the approximation of the solution y_n. The second equation involves the first derivative of the function ρ and evaluates it at four different points. Specifically, we approximate the derivative of the solution y

at t_n by f_n, and we approximate it at three other points, namely t_{n+c_1}, t_{n+c_2}, and t_{n+1}, by f_{n+c_1}, f_{n+c_2}, and f_{n+1}, respectively. In this context, y_{n+j} and f_{n+j} are approximations of the solution $y(t_{n+j})$ and its derivative $f(t_{n+j}, y(t_{n+j}))$, respectively, where j is either 0, c_1, c_2, or 1. The system of equations is solved for the unknowns a_k, and the matrix form of the obtained system is

$$\begin{pmatrix} 1 & t_n & t_n^2 & t_n^3 & t_n^4 \\ 0 & 1 & 2t_n & 3t_n^2 & 4t_n^3 \\ 0 & 1 & 2t_{n+c_1} & 3t_{n+c_1}^2 & 4t_{n+c_1}^3 \\ 0 & 1 & 2t_{n+c_2} & 3t_{n+c_2}^2 & 4t_{n+c_2}^3 \\ 0 & 1 & 2t_{n+1} & 3t_{n+1}^2 & 4t_{n+1}^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} y_n \\ f_n \\ f_{n+c_1} \\ f_{n+c_2} \\ f_{n+1} \end{pmatrix}.$$

Once the values of a_k, $k = 0(1)4$ have been obtained, we can modify the variable t as $t = t_n + zh$, and represent the polynomial in (6) using $\alpha_0(z)$ and $\beta_i(z)$, where $i = 0, c_1, c_2, 1$. Then, the polynomial in (6) may be rewritten as

$$\rho(t_n + zh) = \alpha_0(z) y_n + h\left(\beta_0(z) f_n + \beta_{c_1}(z) f_{n+c_1} + \beta_{c_2}(z) f_{n+c_2} + \beta_1(z) f_{n+1}\right), \tag{8}$$

where $\alpha_0(z) = 1$, $\left\{\beta_i(z)\right\}_{i=0,c_1,c_2,1}$ and the values of $\beta_i(z)$ are dependent on c_1 and c_2. We optimize the local truncation error (LTE) of (8) in order to obtain optimal values of c_1 and c_2 for approximating $y(t_{n+1})$. This LTE is given by

$$L[y(t_{n+1}), h] = \frac{\left(5c_2 + c_1\left(5 - 10c_2\right) - 3\right) h^5 y^{(5)}\left(t_n\right)}{1440} + \mathcal{O}(h^6). \tag{9}$$

By setting the leading term in the LTE (9) to zero, we arrive at the following equation

$$5c_2 + c_1\left(5 - 10c_2\right) - 3 = 0, \tag{10}$$

which we can solve to obtain a relationship between c_1 and c_2. Specifically, we can express c_2 in terms of c_1 as

$$c_2 = \frac{5c_1 - 3}{10c_1 - 5}. \tag{11}$$

It is worth noting that there are infinitely many pairs of values for c_1 and c_2 that satisfy Eq. (10) and the condition $0 < c_1 < c_2 < 1$. However, by optimizing the LTE in (9) for approximating $y(t_{n+1})$, we can choose a specific pair of values for c_1 and c_2 that yields an accurate approximation. In this case, one such optimized pair is given by $c_1 = \frac{1}{3}$ and $c_2 = \frac{4}{5}$.

Substituting $z = 1$ in Eq. (8) yields the formula for approximating the solution $y(t_{n+1})$. This is because $t_{n+1} = t_n + h$, and setting $z = 1$ corresponds to evaluating the polynomial at t_{n+1}. The resulting formula is given as follows:

$$y_{n+1} = y_n + \frac{h}{336}\left(35 f_n + 162 f_{n+c_1} + 125 f_{n+c_2} + 14 f_{n+1}\right). \tag{12}$$

We also proceed to evaluate the polynomial $\rho(t)$ at the points t_{n+c_1}, t_{n+c_2} (that is, taking $z = c_1, c_2$ in (8)) to get the three formulas that form the new OHBT. Here are the remaining formulas:

$$\begin{aligned} y_{n+c_1} &= y_n + \frac{h}{9072}\left(1169 f_n + 2214 f_{n+c_1} - 625 f_{n+c_2} + 266 f_{n+1}\right), \\ y_{n+c_2} &= y_n + \frac{2h}{2625}\left(133 f_n + 648 f_{n+c_1} + 325 f_{n+c_2} - 56 f_{n+1}\right). \end{aligned} \tag{13}$$

Formulas (12)–(13) are the block method that will be used to solve the ordinary differential system in (5).

4. Theoretical analysis

Here, the mathematical properties and theoretical aspects of the OHBT are discussed in detail. The analysis in this section provides a rigorous understanding of the OHBT method and its performance, which is essential for its practical application. The analysis includes the calculation of the local truncation errors (LTEs), the determination of the order of accuracy, and the discussion of stability and convergence properties.

4.1. Local truncation errors, consistency, and convergence of the OHBT

The formulas in Eqs. (12)–(13) can be written as a recurrence relation as follows:

$$\bar{R}\, Y_n = h\, \bar{S}\, Y_n', \tag{14}$$

in which the constant matrices $\bar{R}, \bar{S}$ represent the coefficients of the formulas (12)–(13), and the vectors Y_n and $Y'n$ represent the values of the solution and its derivative at the nodes t_n, t_{n+c_1}, t_{n+c_2}, and t_{n+1}.

To analyze the error of the OHBT method, we define an operator L that is related to the method presented in Eqs. (12)–(13). This operator L takes as input the function $y(t)$ and the stepsize h as follows:

$$L[y(t_n); h] = \sum_{j \in I}\left[\Theta_j\, y\left(t_n + jh\right) - h\Psi_j\, y'\left(t_n + jh\right)\right], \tag{15}$$

Table 1
Local truncation error and order of accuracy for the OHBT.

Formula	Order	Local truncation error
y_{n+c_1}	4	$-\frac{h^5 y^{(5)}(t_n)}{7290} + \mathcal{O}(h^6)$
y_{n+c_2}	4	$\frac{4h^5 y^{(5)}(t_n)}{140625} + \mathcal{O}(h^6)$
y_{n+1}	5	$\frac{h^6 y^{(6)}(t_n)}{108000} + \mathcal{O}(h^7)$

where the terms Θ_j and Ψ_j are vector columns within the matrices $\bar{R}$ and $\bar{S}$, respectively. The set I is defined as $0, c_1, c_2, 1$, which are the indices used in the OHBT method. Assuming that $y(t)$ is sufficiently differentiable, we expand $y(t_n + jh)$ and $y'(t_n + jh)$ in a Taylor series about the point t_n, where j represents the index in the set I. Doing this, we get

$$L[y(t_n); h] = \theta_0 y(t_n) + \theta_1 h y'(t_n) + \theta_2 h^2 y''(t_n) + \cdots + \theta_q h^q y^q(t_n) + \ldots, \tag{16}$$

where

$$\theta_q = \frac{1}{q!}\left[\sum_{j\in I}^{k} j^q \Theta_j - q\sum_{j\in I}^{k} j^{q-1}\Psi_j\right], \quad q = 0, 1, 2, \ldots. \tag{17}$$

By applying the definitions of order and LTEs as provided in [17,31], we can derive the order (p) and expressions of the LTEs for the obtained formulas.

Table 1 presents the LTEs and orders of the obtained formulas. It is important to note that the OHBT is a one-step method; therefore, OHBT is a zero-stable method. Additionally, the method is consistent because the order p is greater than 1. Furthermore, the OHBT is a convergent method due to its zero-stability and consistency.

4.2. Linear-stability analysis

Following the guidelines of the method proposed by [31], we analyze the linear stability of the proposed optimized method. Applying the proposed OHBT to $y' = \nu y$, $Re(\nu) < 0$, we get

$$Y_{n+1} = M(z)Y_n, \quad z = \nu h, \tag{18}$$

where ν is a complex number and $M(z)$ is indeed the stability matrix, which can be expressed as

$$M(z) = \left(C^0 + zE\right)^{-1}\left(C^1 + zD\right), \tag{19}$$

with

$$C^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad C^1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

$$E = \begin{pmatrix} 0 & 0 & 1 + \frac{167}{1296} \\ 0 & 0 & 1 + \frac{38}{375} \\ 0 & 0 & 1 + \frac{5}{48} \end{pmatrix},$$

$$D = \begin{pmatrix} 1 - \frac{41}{168} & \frac{625}{9072} & -\frac{19}{648} \\ -\frac{432}{875} & 1 - \frac{26}{105} & \frac{16}{375} \\ -\frac{27}{56} & -\frac{125}{336} & 1 - \frac{1}{24} \end{pmatrix}.$$

The obtained eigenvalues of the stability matrix $M(z)$ for the OHBT are $\left\{0, 0, \frac{z^3+15z^2+84z+180}{-2z^3+21z^2-96z+180}\right\}$.

We note that the above eigenvalues are essential because the stability of the numerical solution depends on those eigenvalues. Evaluating the stability of the OHBM requires considering the spectral radius, denoted as $\rho(M(z))$, which provides valuable insights into the stability behavior. The region where the absolute value of the spectral radius is less than one, that is, $\mathbb{S} = \{z \in \mathbb{C} : |\rho(M(z))| < 1\}$, represents the resulting region of absolute stability. Fig. 1 visually represents the stability region of the OHBT method, revealing that the entire left half of the complex plane lies within this region. This observation indicates that the OHBT method is A-stable, signifying its ability to solve the differential problem under consideration.

5. Error estimation and control strategy of the OHBT

A variable stepsize formulation (VSSF) for the OHBT will now be presented in this section. Using VSSF, we can obtain a more efficient numerical approximation of the transformed singularly perturbed problem presented in Eq. (5). The introduction of this manuscript emphasizes the importance of changing the stepsize to obtain a more reliable solution to a differential problem. We used

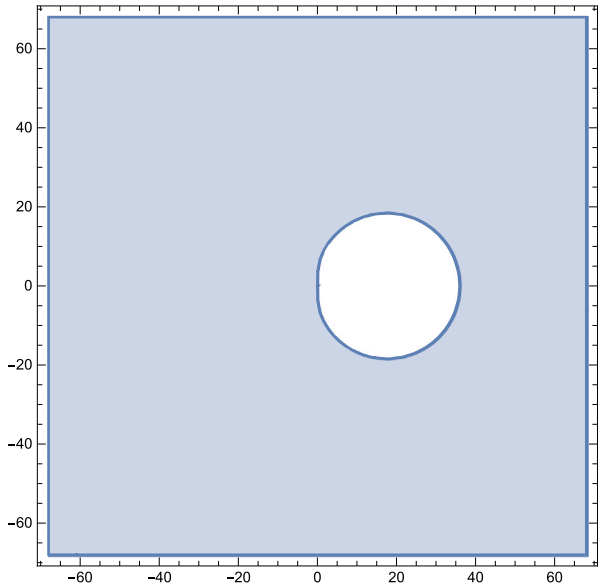

Fig. 1. The region in the complex z-plane where the OHBT method is absolutely stable.

an embedding-like strategy to estimate the local truncation error (LTE) robustly. The lower-order approach used for error estimation of the OHBT is as follows:

$$y^*_{n+1} = y_n + \frac{h}{56}\left(7f_n + 24f_{n+c_1} + 25f_{n+c_2}\right), \tag{20}$$

with $LTE = \frac{h^4 y^{(4)}(t_n)}{1080} + O\left(h^5\right)$. Thus, the local error is estimated through the quotient

$$EST = \frac{\|y^*_{n+1} - y_{n+1}\|}{\left(1 + \|y_{n+1}\|\right)}. \tag{21}$$

Before continuing our calculations, it is necessary to verify that the absolute value of the estimated error (EST) is equal to or less than the specified relative tolerances (RET). The results are accepted if $EST \leq$ RET, while the stepsize is increased to minimize the computational cost. If $EST >$ RET, the results are rejected, and the stepsize is adjusted using the following stepsize selection strategy to improve the accuracy

$$h_{vary} = \eta\, h_{old}\left(\frac{RET}{\|EST\|}\right)^{1/(p+1)}. \tag{22}$$

In Eq. (22), $p = 3$ represents the order of the lower order approach in (20), and we prevent failed steps by using a safety factor η. In our numerical experiment, we set η to 0.90, and we initialized the stepsize that will be changed by the algorithm if needed using the method described earlier to ensure accuracy and efficiency.

6. Implementation of the proposed method

Eq. (1) is discretized using approximations for y_x and y_{xx} at (x_i, t) as given in Section 2. Then, we rewrite the system in (12)–(13) for the discretized problem obtained from (5) as $\boldsymbol{F(y) = 0}$, where the unknowns are represented by the following vector $\tilde{\mathbf{Y}}$

$$\tilde{\mathbf{Y}} = \left(y_{1,n+c_1}, y_{2,n+c_1}, \cdots, y_{N-1,n+c_1}, y_{1,n+c_2}, y_{2,n+c_2}, \cdots, y_{N-1,n+c_2}, y_{1,n+1}, y_{2,n+1}, \cdots, y_{N-1,n+1}\right).$$

We use Newton's method (NM) to solve the obtained nonlinear equations. The NM is an iterative method that starts with an initial guess of the solution and then updates the guess by computing the solution of a linear system at each iteration until convergence. For the OHBT scheme, the NM takes the form of

$$\tilde{\mathbf{Y}}^{\mathbf{i+1}} = \tilde{\mathbf{Y}}^{\mathbf{i}} - \left(\mathbf{J_0}\right)^{-1}\mathbf{F}^{\mathbf{i}},$$

where $\tilde{\mathbf{Y}}^{\mathbf{i}}$ is the current guess of the solution, $\mathbf{F}^{\mathbf{i}}$ is the corresponding nonlinear function evaluated at $\tilde{\mathbf{Y}}^{\mathbf{i}}$, and $\mathbf{J_0}$ is the frozen Jacobian matrix of $\mathbf{F}$ evaluated at $\tilde{\mathbf{Y}}^{\mathbf{i}}$. To obtain a starting point for the NM iterations, we can use an approximation of the solution based on the values of the previous time step. In the OHBT scheme, we use the values of y_n and f_n (which are known) to compute the starting values for the unknowns in each iteration, given by $y_{n+j} = y_n + (jh)f_n$, where h is the time step size and $j = c_1, c_2, 1$ correspond to the three different sets of unknowns in the system. These starting values help speed up the convergence of the NM iterations,

reducing the computational cost of solving the system at each time step. Following is an outline of the steps that are taken for the proposed method to be used in integrating problem (5):

1. Initialize the algorithm by setting the initial stepsize (h), the initial condition (t_0, y_0), the number of spatial subintervals N, and the end-point of the integration interval T.
2. Define the function $f(t, y(t))$ and initialize t_n and y_n.
3. Check if the current value of t_n is greater than or equal to the end-point T. If true, then exit the algorithm.
4. If $t_n + h$ is greater than T, reduce the stepsize (h) to ensure that the next step does not exceed the end-point.
5. While t_n is less than T, solve the system of equations in (12)–(13) using the current step size h_{old} to obtain the value of y_{n+1}.
6. Compute an estimate of the error using $EST = \frac{\|y^*_{n+1} - y_{n+1}\|}{(1+\|y_{n+1}\|)}$.
7. If EST is less than or equal to the user-defined relative tolerance RET, accept the results and increase the stepsize (h) by a factor of 2. Set t_n to $t_n + h$, set y_n to y_{n+1}, and go to step 3.
8. If EST is greater than RET, reject the results, compute a new stepsize (h) using Eq. (22), and go back to step 5.

The above algorithm iteratively computes an approximate solution on a discrete mesh by adapting the stepsize based on the error estimate. The algorithm increases the stepsize if the error is small and reduces the stepsize if the error is large. The algorithm terminates when the end-point of the integration interval is reached. The user-defined relative tolerance RET controls the accuracy of the solution. The smaller the RET, the more accurate the solution, but the longer the algorithm takes to compute. The stepsize (h) is critical to the accuracy and efficiency of the algorithm. A large stepsize may result in inaccurate solutions, while a small stepsize may result in excessive computational time. Therefore, the algorithm uses an adaptive stepsize for better accuracy and efficiency.

7. Numerical experiments

Some test problems are presented here to evaluate the effectiveness of the proposed OHBT method for solving the problem under consideration. These problems are used to compare the numerical results obtained by the OHBT method with those obtained by other existing numerical methods. By comparing the OHBT with other methods, one can determine the accuracy and reliability of the proposed method. In order to compare the accuracy of different methods, the maximum absolute error (MAE) is calculated using the following formula:

$$MAE = \max_{i=0,\ldots,N, j=0,\ldots,N} \left\| y(x_i, t_j) - Y(x_i, t_j) \right\|_\infty ,$$

where $y(x_i, t)$ represents the reference solution, and $Y(x_i, t_j)$ is the solution obtained by the OHBT method at each time point t_j.

In the Tables and Figures presented in this paper, we use the following abbreviations to represent different methods and performance measures. The optimized hybrid block technique proposed in this manuscript is denoted by OHBT. The implicit Gauss–Legendre method of order six formulated in variable stepsize is represented by G6P and is cited from [32]. The uniformly convergent B-spline method is denoted by UCBSM and is described in [4]. The higher-order uniformly convergent method is referred to as HOMCM and is reported in [33].

We also use various performance measures to evaluate the accuracy and efficiency of the methods. The number of steps taken by the solver is denoted by NOS, while the number of rejected steps is represented by RS. The number of Newton iterations used is denoted by NI, and the total number of Jacobian evaluations is denoted by TNJ. The total number of function evaluations is represented by FE. The initial step size is denoted by h_{ini}. The computational time taken by the solver is denoted by CT and measured in seconds. Finally, we also compare our proposed method OHBT with the built-in package solver in Mathematica, which we refer to as NDSolve.

7.1. Example 1

Consider the following SPPCDP described in [4]

$$\begin{aligned} &\epsilon y_{xx} + x^p y_x - y_t - y = x^2 - 1\,, (x,t) \in \mathcal{D} = [0,1] \times [0,1], \\ &y(0,t) = 1 + t^2,\ y(1,t) = 0,\ y(x,0) = (1-x)^2. \end{aligned} \tag{23}$$

The problem has boundary conditions specified at $x = 0$ and $x = 1$, and an initial condition specified at $t = 0$. However, the exact solution to this problem is not known.

7.2. Example 2

Consider the SPPCDP discussed in [4]

$$\begin{aligned} &\epsilon y_{xx} + x^p y_x - y_t - (x+p)y = p(x^2-1)\exp(-t)\,, (x,t) \in \mathcal{D} = [0,1] \times [0,1], \\ &y(0,t) = 1 + t^2,\ y(1,t) = 0,\ y(x,0) = (1-x)^2\,. \end{aligned} \tag{24}$$

It should be noted that no exact solution has been found for the above problem.

Table 2
Comparison of numerical results for Problem (23) with $h_{ini} = 10^{-1}, \epsilon = 2^{-8}, p = 1, N = 29$.

RET	Methods	NOS	RS	NI	FE	TNJ	CT	MAE
10^{-2}	OHBT	3	0	5	12	3	0.3125	6.6657×10^{-5}
10^{-2}	G6P	5	0	9	30	5	1.1406	5.6038×10^{-3}
10^{-4}	OHBT	4	0	8	16	4	0.4219	9.3635×10^{-6}
10^{-4}	G6P	21	3	48	126	24	5.7656	8.3356×10^{-4}
10^{-6}	OHBT	10	1	22	40	11	1.1563	1.2579×10^{-7}
10^{-6}	G6P	215	5	440	1290	220	53.1250	1.0650×10^{-4}

Table 3
Comparison of numerical results for Problem (23) with $h_{ini} = 10^{-2}, \epsilon = 2^{-16}, p = 1, N = 29$.

RET	p	Methods	NOS	RS	NI	FE	TNJ	CT	MAE
10^{-5}	2	OHBT	8	0	16	32	8	0.8594	7.3002×10^{-6}
10^{-5}	2	G6P	81	0	162	486	81	20.8750	3.4824×10^{-4}
10^{-5}	6	OHBT	10	0	20	40	10	1.0625	4.9189×10^{-6}
10^{-5}	6	G6P	103	1	208	618	104	26.7031	3.5243×10^{-4}

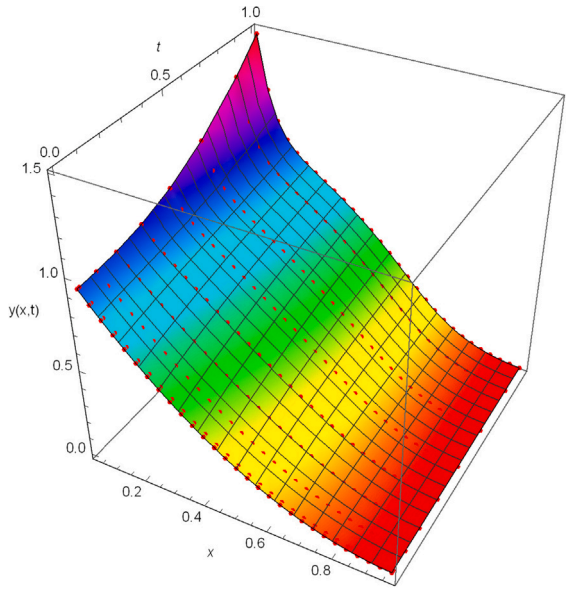

Fig. 2. Surface solution of Problem (23) obtained with NDSolve and the approximate solution obtained with the OHBT method (red dots) using $h_{ini} = 10^{-2}, \epsilon = 2^{-7}, p = 3$.

Table 4
Comparison of numerical results for Problem (24) with $h_{ini} = 10^{-1}, \epsilon = 2^{-10}, p = 1, N = 29$.

RET	Methods	NOS	RS	NI	FE	TNJ	CT	MAE
10^{-2}	OHBT	2	0	3	8	2	0.2500	2.3303×10^{-4}
10^{-2}	G6P	5	0	10	30	5	1.2969	7.4805×10^{-3}
10^{-4}	OHBT	4	0	8	16	4	0.4219	7.2542×10^{-6}
10^{-4}	G6P	26	0	60	156	30	7.7969	1.1184×10^{-3}
10^{-6}	OHBT	9	0	18	36	9	1.0625	1.2934×10^{-7}
10^{-6}	G6P	254	6	520	1524	260	68.1563	1.1290×10^{-4}

7.3. Discussion of results

The numerical results obtained using the suggested OHBT approach are compared to those obtained using the G6P, UCBSM, and HOMCM methods. We used NDSolve to provide the reference solution in order to determine MAE for numerical Experiments 1 and 2. Notably, NDSolve is a numerical method for solving ordinary or partial differential equations implemented in Mathematica. Tables 2–5 show that the numerical results obtained with our current approach is more accurate than the G6P method. The reported MAEs for Problems (23)–(24) with $NOS = 128, p = 1, \epsilon = 2^{-8}$ and $NOS = 128, p = 1, \epsilon = 2^{-10}$ for the B-spline based method in [4]

Table 5
Comparison of numerical results for Problem (24) with $h_{ini} = 10^{-2}, \epsilon = 2^{-20}, N = 29$.

RET	p	Methods	NOS	RS	NI	FE	TNJ	CT	MAE
10^{-5}	6	OHBT	12	0	26	48	13	1.4063	6.1794×10^{-7}
10^{-5}	6	G6P	22	6	456	1332	228	59.4844	3.7334×10^{-4}
10^{-5}	10	OHBT	13	0	30	52	15	1.6563	1.5295×10^{-7}
10^{-5}	10	G6P	271	6	554	1626	277	77.0000	3.9088×10^{-4}

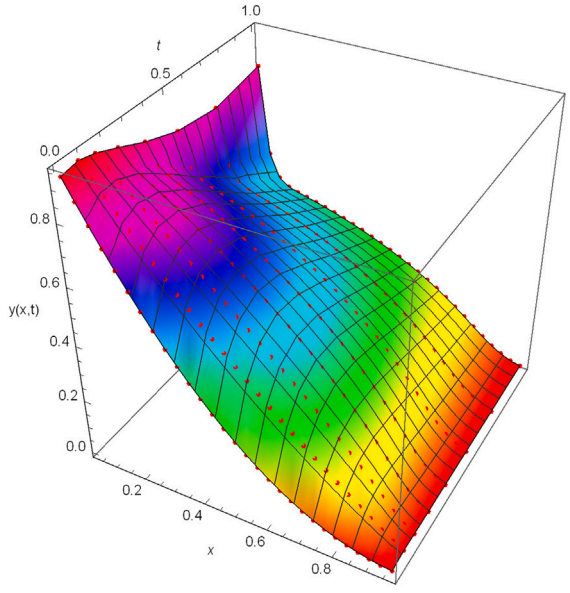

Fig. 3. Surface solution of Problem (24) obtained with NDSolve and the approximate solution obtained with the OHBT method (red dots) using $h_{ini} = 10^{-1}, \epsilon = 2^{-8}, p = 6, N = 29$.

are 3.96300×10^{-5} and 5.4298×10^{-5}, respectively, while for the OHNT with $NOS = 10, p = 1, \epsilon = 2^{-8}$ and $NOS = 9, p = 1, \epsilon = 2^{-10}$ are 1.2579×10^{-7} and 1.2934×10^{-7}, respectively, confirming that OHNT method performs better than the B-spline technique in [4].

In addition, from Tables 3 and 5, we can see the MAEs for Problems (23)–(24) with different values of p (specifically $p = 2$ and $p = 6$). We emphasize that the numerical techniques presented in [4,33], using the same values of p, report MAEs of 4.7157×10^{-5}, 2.2990×10^{-4} and 4.3795×10^{-4}, 8.4190×10^{-4}, respectively, for Problem (23), while for the OHNT method, the obtained MAEs for the same p values are 7.3002×10^{-6}, 4.9189×10^{-6}, respectively, indicating the superior performance of the OHBT method. Figs. 2–3 display surface plots of the proposed OHBT method and NDSolve on Problems (23)–(24) to demonstrate the good agreement between the OHBT method and NDSolve solutions.

8. Conclusion

This paper introduces a new approach to solve a singularly perturbed parabolic convection–diffusion problem. The method is called the optimized hybrid block technique (OHBT), and we present a formulation with variable stepsizes to improved accuracy. To evaluate the effectiveness of OHBT, we compare its results with those obtained using other existing numerical methods. The comparison is based on the numerical solutions of the problem in Eqs. (1)–(4). The reported numerical solutions in Tables 2–5 and Figs. 2–3 indicate that the proposed OHBT outperforms different existing numerical techniques utilized for comparison. Future research work could investigate the possibility of solving the time-dependent partial differential of Lane–Emden–Fowler equations using the type of pairwise methods reported in [34,35].

Declaration of competing interest

We declare that the work presented has not been published previously, and it is not under consideration for publication elsewhere, that authors approve its publication and that it will not be published elsewhere in the same form, in English or any other language, including electronically without the written consent of the copyright holder.

We confirm that the authors have no conflict of interest.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is supported by Provincia Autonoma di Bolzano/Alto Adige - Ripartizione Innovazione, Ricerca, Universita e Musei (contract nr. 19/34).

References

[1] Hanks TC. Model relating heat-flow values near, and vertical velocities of mass transport beneath, oceanic rises. J Geophys Res 1971;76:537–44.
[2] Cole JD. On a quasi-linear parabolic equation occurring in aerodynamics. Q Appl Math 1951;9:225–36.
[3] Polak SJ, Den Heijer C, Schilders WHA, Markowich P. Semiconductor device modeling from the numerical point of view. Int J Numer Methods Eng 1987;24:763–838.
[4] Singh S, Kumar D, Ramos H. A uniformly convergent quadratic B-spline based scheme for singularly perturbed degenerate parabolic problems. Math Comput Simulation 2022;195:88–106.
[5] Schlichting H, Gersten K. Boundary-layer theory. New York: MacGraw Hill; 1979.
[6] Roos HG, Stynes L. Tobiska M. Robust numerical methods for singularly perturbed differential equations: convection-diffusion-reaction and flow problems. Berlin: Springer; 2008.
[7] Shishkin GI, Shishkina LP. Difference methods for singular perturbation problems. New York: CRC Press; 2008.
[8] Kumar D. A parameter-uniform method for singularly perturbed turning point problems exhibiting interior or twin boundary layers. Int J Comput Math 2019;96:865–82.
[9] Kumar S, Kumar M. Parameter-robust numerical method for a system of singularly perturbed initial value problems. Numer Algorithms 2012;59(2):185–95.
[10] Constantinou P, Xenophontos C. Finite element analysis of an exponentially graded mesh for singularly perturbed problems comput. Methods Appl Math 2015;15:135–43.
[11] Kumar S. Layer-adapted methods for quasilinear singularly perturbed delay differential problems. Appl Math Comput 2014;233:214–21.
[12] Rufai MA. An efficient third derivative hybrid block technique for the solution of second-order BVPs. Mathematics 2022;10(19):3692.
[13] Selvi PA, Ramanujam N. A parameter uniform difference scheme for singularly perturbed parabolic delay differential equation with Robin type boundary condition. Appl Math Comput 2017;296:101–15.
[14] Qiu Y, Sloan D. Analysis of difference approximations to a singularly perturbed two-point boundary value problem on an adaptively generated grid. J Comput Appl Math 1999;101(12):1–25.
[15] Kumar S, Sumit, Vigo-Aguiar J. A high order convergent numerical method for singularly perturbed time dependent problems using mesh equidistribution. Math Comput Simulation 2022;199:287–306.
[16] Alam MP, Kumar D, Khan A. Trigonometric quintic B-spline collocation method for singularly perturbed turning point boundary value problems. Int J Comput Math 2020;98:1029–48.
[17] Duromola MK, Momoh AL, Rufai MA, Animasaun IL. Insight into 2-step continuous block method for solving mixture model and SIR model. Int J Comput Sci Math 2022;14(4):347–56.
[18] Arora G, Joshi V. A computational approach for solution of one dimensional parabolic partial differential equation with application in biological processes. Ain Shams Eng J 2018;9(4):1141–50.
[19] Anastassi ZA, Kosti AA, Rufai MA. A parametric method optimised for the solution of the (2+1)- dimensional nonlinear Schrödinger equation. Mathematics 2023;11(3):609.
[20] Gelu FW, Duressa GF. Parameter-uniform numerical scheme for singularly perturbed parabolic convection–diffusion robin type problems with a boundary turning point. Results Appl Math 2022;15:100324.
[21] Hailu WS, Duressa GF. Accelerated parameter-uniform numerical method for singularly perturbed parabolic convection–diffusion problems with a large negative shift and integral boundary condition. Results Appl Math 2023;18:100364.
[22] Rufai MA, Mazzia F, Ramos H. An adaptive optimized Nyström method for second-order IVPs. Math Methods Appl Sci 2023;46(6):7543–56.
[23] Rufai MA, Tran T, Anastassi ZA. A variable step-size implementation of the hybrid Nyström method for integrating Hamiltonian and stiff differential systems. Comput Appl Math 2023;42:156.
[24] Viscor M, Stynes M. A robust finite difference method for a singularly perturbed degenerate parabolic problem, part I. Int J Numer Anal Model 2010;7:549–66.
[25] Li J. Convergence analysis of finite element methods for singularly perturbed problems. Comput Math Appl 2000;40(6–7):735–45.
[26] Li J. Optimal uniform convergence analysis for a two-dimensional parabolic problem with two small parameters. Int J Numer Anal Model 2005;2(1):107–26.
[27] Ramos H, Vigo-Aguiar J, Natesan S, Queiruga MA. Numerical solution of nonlinear singularly perturbed problems on nonuniform meshes by using a non-standard algorithm. J Math Chem 2010;48:38–54.
[28] Ayele MA, Tiruneh AA, Derese GA. Fitted cubic spline scheme for two-parameter singularly perturbed time-delay parabolic problems. Results Appl Math 2023;18:100361.
[29] Rufai MA, Ramos H. Numerical solution of Bratu's and related problems using a third derivative hybrid block method. Comput Appl Math 2020;39(4):322.
[30] Rufai MA, Ramos H. Solving SIVPs of Lane–Emden–Fowler type using a pair of optimized Nyström methods with a variable step size. Mathematics 2023;11(6):1535.
[31] Ramos H, Rufai MA. An adaptive one-point second-derivative Lobatto-type method for solving efficiently differential systems. Int J Comput Math 2022;99(8):1687–705.
[32] Butcher JC. Numerical methods for ordinary differential equations. John Wiley and Sons; 2008.
[33] Yadav S, Rai P, Sharma KK. A higher order uniformly convergent method for singularly perturbed parabolic turning point problems. Numer Methods Partial Differential Equations 2020;36:342–68.
[34] Ramos H, Rufai MA, Carpentieri B. A pair of optimized Nyström methods with symmetric hybrid points for the numerical solution of second-order singular boundary value problems. Symmetry 2023;15(9):1720.
[35] Rufai MA, Ramos H. Solving third-order Lane–Emden–Fowler equations using a variable step-size formulation of a pair of block methods. Comput Appl Math 2022;420:114776.