British Journal of Mathematics & Computer Science
3(3): 397-414, 2013

SCIENCEDOMAIN *international*
www.sciencedomain.org

Numerical Integration with Exponential Fitting Factor for Singularly Perturbed Two Point Boundary Value Problems

K. Madhu Latha[1], K. Phaneendra[2*] and Y. N. Reddy[1]

[1]*Department of Mathematics, National Institute of Technology, Warangal, India.*
[2]*Department of Mathematics, Kakatiya Institute of Technology & Science, Warangal, India.*

Research Article

Received: 22 March 2013
Accepted: 17 May 2013
Published: 15 June 2013

Abstract

In this paper, we discuss the numerical integration with exponential fitting factor for singularly perturbed two-point boundary value problems. It is based on the fact that: the given SPTPBVP is replaced by an asymptotically equivalent delay differential equation. Then, numerical integration with exponential fitting factor is employed to obtain a tridiagonal system which is solved efficiently by Thomas algorithm. We discussed convergence analysis of the method. Model examples are solved and the numerical results are compared with exact solution.

Keywords: Singular perturbation problem, boundary layer, delay differential equation, fitting factor, trapezoidal rule.

1 Introduction

Singular perturbation problems arise in various fields of engineering and applied sciences such as fluid dynamics, electrical networks, and many other areas. Typical examples of Singular Perturbation Problems include Navier-Stokes equation of fluid at high Reynolds number, heat transport problem with Peclet numbers, magneto-hydrodynamics duct problems with Hartman number. A differential equation with a small positive parameter multiplying the highest derivative term is generally called the *Singular Perturbation Problem.* The solution of singular perturbation problem exhibits boundary layers. A boundary layer is a narrow region in which solution of the problem changes rapidly. For these problems, the existing numerical methods produce good results only if we take $h << \varepsilon$. But this is costly and time consuming process. If we take $h \geq \varepsilon$, the existing numerical methods produce oscillatory solution and pollute the solution in the entire interval, because of the boundary layer behavior. For a detailed theoretical and analytical discussion on this topic, one may refer to the references [1-11].

**Corresponding author: kollojuphaneendra@yahoo.co.in;*

The numerical treatment of singular perturbation problems is far from trivial because of the boundary layer behaviour of the solution. However, the area of singular perturbation problems is a field of increasing interest to applied mathematicians. Engineers and applied scientists want more efficient and simpler computational techniques which can be readily implemented on computer for solving both singular perturbation problems. Recently, Soujanya et al. [12] presented a numerical solution of singular perturbation problem using deviating argument and exponential fitting factor.

In this paper, we modify/improve the idea/concept given in [12]. Here, we present the numerical integration with exponential fitting factor for singularly perturbed two-point boundary value problems. It is based on the fact that: the given SPTPBVP is replaced by an asymptotically equivalent delay differential equation. Then, numerical integration with exponential fitting factor is employed to obtain a tridiagonal system which is solved efficiently by Thomas algorithm. We also discussed convergence analysis of the method. Model examples are solved and the numerical results are compared with exact solution.

2 Description of the Method

To describe our method, let us consider singularly perturbed singular boundary value problems of the form

$$\varepsilon y''(x) + a(x)y'(x) + b(x)y(x) = f(x), \quad 0 \le x \le 1, \tag{1}$$

with boundary conditions

$$y(0) = \alpha \tag{2a}$$

and

$$y(1) = \beta \tag{2b}$$

where $0 < \varepsilon << 1$, $a(x), b(x), f(x)$ are bounded continuous functions in [0, 1] and α, β are finite constants. If we assume $a(x) \ge M > 0$ throughout the interval [0, 1], where M is positive constant, then the boundary layer will be in the neighbourhood of $x = 0$ and if $a(x) \le M < 0$ throughout the interval [0, 1], where M is negative constant, then the boundary layer will be in the neighbourhood of left end point i.e., at $x = 1$.

2.1 Left End Layer Problem

By using Taylor series expansion in the neighbourhood of the point x and the small positive deviating argument $\sqrt{\varepsilon}$, we have

$$y\left(x - \sqrt{\varepsilon}\right) = y(x) - \sqrt{\varepsilon}\, y'(x) + \frac{\varepsilon}{2} y''(x)$$

$$y''(x) = \frac{2y\left(x-\sqrt{\varepsilon}\right) - 2y(x) + 2\sqrt{\varepsilon}\,y'(x)}{\varepsilon} \tag{3}$$

and consequently, equation (1) is replaced by the following first order delay differential equation:

$$y'(x) = p(x)y(x-\sqrt{\varepsilon}) + q(x)y(x) + r(x)\text{, for } \sqrt{\varepsilon} \le x \le 1 \tag{4}$$

where

$$p(x) = \frac{-2}{2\sqrt{\varepsilon}+a(x)},\ q(x) = \frac{2-b(x)}{2\sqrt{\varepsilon}+a(x)} \text{ and } r(x) = \frac{f(x)}{2\sqrt{\varepsilon}+a(x)}$$

The transition from equation (1) to equation (4) is allowed, because of the condition that $\sqrt{\varepsilon}$ is small. The validity of this transition can be found in El'sgol'ts and Norkin [4]. This replacement is significant from the computational point of view.

Now we divide the interval [0, 1] into N equal subintervals of mesh size $h = 1/N$ so that $x_i = ih,\ i = 0, 1, 2, \ldots, N$.

Here, for consolidations our ideas of the method, we assume that $a(x)$ and $b(x)$ are constants whereas in [12] authors considered $a(x)$, $b(x)$ and $f(x)$ as constants. Hence, here $p(x)$ and $q(x)$ are only the constants.

Rewriting the equation (4) as $y'(x) - qy(x) = py(x-\sqrt{\varepsilon}) + r(x)$

We then apply an integrating factor e^{-qx}, producing (as in Brian J. McCartin [13])

$$\frac{d}{dx}\left[e^{-qx}y(x)\right] = e^{-qx}\left[py(x-\sqrt{\varepsilon}) + r(x)\right]$$

Next, integrating from x_i to x_{i+1}, we get

$$e^{-qx_{i+1}}y_{i+1} - e^{-qx_i}y_i = \int_{x_i}^{x_{i+1}} e^{-qx}py(x-\sqrt{\varepsilon})dx + \int_{x_i}^{x_{i+1}} e^{-qx}r(x)dx$$

Using the Trapezoidal rule to evaluate the integrals and simplifying, we get

$$y_{i+1} = e^{qh}y_i + \frac{hp}{2}\left(e^{qh}y(x_i-\sqrt{\varepsilon}) + y(x_{i+1}-\sqrt{\varepsilon})\right) + \frac{h}{2}\left(e^{qh}r_i + r_{i+1}\right) \tag{5}$$

Approximating $y'(x)$ by linear interpolation, we get

$$y(x_i - \sqrt{\varepsilon}) \approx \left(1 - \frac{\sqrt{\varepsilon}}{h}\right) y_i + \frac{\sqrt{\varepsilon}}{h} y_{i-1} \tag{6}$$

$$y(x_{i+1} - \sqrt{\varepsilon}) \approx \left(1 - \frac{\sqrt{\varepsilon}}{h}\right) y_{i+1} + \frac{\sqrt{\varepsilon}}{h} y_i \tag{7}$$

Substituting (6), (7) in equation (5) and rearranging the terms we get the following three term relation

$$\left(\frac{-e^{qh}\sqrt{\varepsilon}\,p}{2}\right) y_{i-1} - \left(e^{qh} + \frac{e^{qh}ph}{2}\left(1 - \frac{\sqrt{\varepsilon}}{h}\right) + \frac{\sqrt{\varepsilon}\,p}{2}\right) y_i + \left(1 - \frac{ph}{2}\left(1 - \frac{\sqrt{\varepsilon}}{h}\right)\right) y_{i+1} = \frac{h}{2}\left(e^{qh} r_i + r_{i+1}\right)$$

for $i = 1,2,\ldots,N\text{-}1$.

The above relation can be written as

$$E_i y_{i-1} - F_i y_i + G_i y_{i+1} = H_i, \text{ for } i = 1, 2, \ldots, N\text{-}1 \tag{8}$$

where

$$E_i = \frac{-e^{qh}\sqrt{\varepsilon}\,p}{2}, \; F_i = e^{qh} + \frac{e^{qh}ph}{2}\left(1 - \frac{\sqrt{\varepsilon}}{h}\right) + \frac{\sqrt{\varepsilon}\,p}{2}$$

$$G_i = 1 - \frac{ph}{2}\left(1 - \frac{\sqrt{\varepsilon}}{h}\right), \; H_i = \frac{h}{2}\left(e^{qh} r_i + r_{i+1}\right)$$

Equation (8) is a tridiagonal system.

2.2 Right End Layer Problem

By using the small positive deviating argument $\sqrt{\varepsilon}$ and using Taylor series expansion in the neighbourhood of the point x, we have

$$y\left(x + \sqrt{\varepsilon}\right) = y(x) + \sqrt{\varepsilon}\, y'(x) + \frac{\varepsilon}{2} y''(x)$$

$$y''(x) = \frac{2y\left(x + \sqrt{\varepsilon}\right) - 2y(x) - 2\sqrt{\varepsilon}\, y'(x)}{\varepsilon} \tag{9}$$

and consequently, equation (1) is replaced by the following first order delay differential equation:

$$y'(x) = p(x)y(x + \sqrt{\varepsilon}) + q(x)y(x) + r(x), \text{ for } \sqrt{\varepsilon} \le x \le 1 \tag{10}$$

where

$$p(x) = \frac{-2}{2\sqrt{\varepsilon} + a(x)}, \; q(x) = \frac{2 - b(x)}{2\sqrt{\varepsilon} + a(x)} \text{ and } r(x) = \frac{f(x)}{2\sqrt{\varepsilon} + a(x)}$$

We divide the interval [0, 1] into N equal subintervals of mesh size h = 1/N so that $x_i = ih, \; i = 0,$ 1, 2,…, N.

Here, for consolidations our ideas of the method, we assume that $a(x)$ and $b(x)$ are constants. Hence, $p(x)$ and $q(x)$ are constant.

Rearranging equation (10) as $y'(x) - qy(x) = py(x + \sqrt{\varepsilon}) + r(x)$

We then apply an integrating factor e^{-qx}, producing

$$\frac{d}{dx}\left[e^{-qx} y(x)\right] = e^{-qx}\left[py(x + \sqrt{\varepsilon}) + r(x)\right]$$

Now, integrating from x_{i-1} to x_i, we get

$$e^{-qx_i} y_i - e^{-qx_{i-1}} y_{i-1} = \int_{x_{i-1}}^{x_i} e^{-qx} py(x + \sqrt{\varepsilon}) dx + \int_{x_{i-1}}^{x_i} e^{-qx} r(x) dx$$

By making use of Trapezoidal rule to evaluate the integrals and simplifying, we get

$$y_i = e^{qh} y_{i-1} + \frac{hp}{2}\left(y(x_i + \sqrt{\varepsilon}) + e^{qh} y(x_{i-1} + \sqrt{\varepsilon})\right) + \frac{h}{2}\left(r_i + e^{qh} r_{i-1}\right) \tag{11}$$

Approximating $y'(x)$ by linear interpolation, we get

$$y(x_i + \sqrt{\varepsilon}) \approx \left(1 - \frac{\sqrt{\varepsilon}}{h}\right) y_i + \frac{\sqrt{\varepsilon}}{h} y_{i+1} \tag{12}$$

$$y(x_{i-1} + \sqrt{\varepsilon}) \approx \left(1 - \frac{\sqrt{\varepsilon}}{h}\right) y_{i-1} + \frac{\sqrt{\varepsilon}}{h} y_i \tag{13}$$

Substituting (12), (13) in equation (11) and rearranging the terms we get the following three term relation

$$\left(-e^{qh} - \frac{e^{qh} hp}{2}\left(1 - \frac{\sqrt{\varepsilon}}{h}\right)\right) y_{i-1} - \left(-1 + \frac{hp}{2} - \frac{\sqrt{\varepsilon}\, p}{2} + \frac{\sqrt{\varepsilon}\, p e^{qh}}{2}\right) y_i + \left(\frac{-\sqrt{\varepsilon}\, p}{2}\right) y_{i+1} = \frac{h}{2}\left(r_i + e^{qh} r_{i-1}\right)$$

for i = 1,2,…,N-1.

The above relation can be written as

$$E_i y_{i-1} - F_i y_i + G_i y_{i+1} = H_i \text{, for i = 1, 2,…, } N\text{-1.} \tag{14}$$

where

$$E_i = -e^{qh} - \frac{e^{qh}hp}{2}\left(1-\frac{\sqrt{\varepsilon}}{h}\right),\ F_i = -1 + \frac{hp}{2} - \frac{\sqrt{\varepsilon}\,p}{2} + \frac{\sqrt{\varepsilon}\,pe^{qh}}{2}$$

$$G_i = \frac{-\sqrt{\varepsilon}\,p}{2},\ H_i = \frac{h}{2}\left(r_i + e^{qh} r_{i-1}\right)$$

Equation (8) is a tridiagonal system.

We solve the tridiagonal system (8) or (14) by using an efficient Thomas algorithm.

3 Convergence Analysis

Writing the tridiagonal system (8) in matrix-vector form, we get

$$AY = C \tag{15}$$

in which $A = (m_{i\,j}),\ 1 \le i, j \le n\text{-}1$ is a tridiagonal matrix of order N-1 , with

$$m_{i\,i+1} = 1 - \frac{ph}{2}\left(1-\frac{\sqrt{\varepsilon}}{h}\right),$$

$$m_{i\,i} = e^{qh} + \frac{e^{qh}ph}{2}\left(1-\frac{\sqrt{\varepsilon}}{h}\right) + \frac{\sqrt{\varepsilon}\,p}{2},$$

$$m_{i\,i\text{-}1} = \frac{-e^{qh}\sqrt{\varepsilon}\,p}{2}{}_i$$

and $C = (d_i)$ is a column vector with $d_\text{i} = \frac{h}{2}\left(e^{qh} r_i + r_{i+1}\right)$, where $i = 1\ (1)\ N\text{-}1$

with local truncation error $T_\text{i}(h_i) = h\left[\frac{-\varepsilon\, y_i'''}{2\sqrt{\varepsilon}+a}\right] + o(h^2)$ (16)

We also have

$$A\overline{Y} - T(h) = C \tag{17}$$

where $\overline{Y} = \left(\overline{y_0}, \overline{y_1}, \overline{y_2}, \ldots\ldots\ldots\ldots\ldots, \overline{y_N}\right)^t$ denotes the actual solution and

$T(h) = (T_1(h_0), T_2(h_1), \ldots\ldots\ldots, T_N(h_N))^t$ is the local truncation error.

From (15) and (17), we get

$$A\left(\overline{Y} - Y\right) = T(h) \tag{18}$$

Thus the error equation is $\quad AE = T(h) \quad$ (19)

where $E = \overline{Y} - Y = (e_0, e_1, e_2, \ldots\ldots\ldots\ldots e_N)^t$.

Clearly, we have

$$S_1 = \sum_{j=1}^{N-1} m_{1\,\mathrm{j}} = \left(1 + \frac{p\sqrt{\varepsilon}}{2}\right) - h\left(\frac{pq\sqrt{\varepsilon}}{2}\right) + \frac{h^2}{2}\left(pq - pq^2\sqrt{\varepsilon}\right) + o(h^3)$$

$$S_i = \sum_{j=1}^{N-1} m_{i\,\mathrm{j}} = 2 + h\left[q\left(1 - p\sqrt{\varepsilon}\right)\right] + o(h^2) = 2 + o(h) = B_0 \quad \text{where } B_0 = 2, \text{ for } i = 2,3,.., N-2$$

$$S_{N-1} = \sum_{j=1}^{N-1} m_{N-1\,\mathrm{j}} = \left(1 - p\sqrt{\varepsilon}\right) + h\left(q\left(1 - p\sqrt{\varepsilon}\right) + \frac{p}{2}\right) + o(h^2)$$

We can choose *h* sufficiently small so that the matrix *A* is irreducible and monotone. It follows that A^{-1} exists and its elements are non negative.

Hence from Eq. (19), we get $\quad E = A^{-1}T(h) \quad$ (20)

Also, from the theory of matrices we have

$$\sum_{i=1}^{N-1} \overline{m}_{\mathrm{k,i}}\, S_{\mathrm{i}} = 1\,, \quad k = 1(1)\, N\text{ -}1 \tag{21}$$

where $\overline{m}_{k,i}$ is (k,i) element of the matrix A^{-1}, therefore

$$\sum_{i=1}^{N-1} \bar{m}_{k,i} \le \frac{1}{\min\limits_{1\le i\le N-1} S_i} = \frac{1}{B_0} \le \frac{1}{2} \tag{22}$$

From (16), (20) and (22), we get

$$e_j = \sum_{i=1}^{N-1} \overline{m}_{k.i}\, T_i(h), \quad j = 1\ (1)\ \ N-1$$

which implies

$$e_j \le \frac{kh}{2}\ , \tag{23}$$

where *k* is a constant independent of *h*, that is $k = \dfrac{-\varepsilon\, y''}{2\sqrt{\varepsilon} + a}$

Therefore, $$\|E_i\| = o(h)$$

i.e., our method gives a first order convergent for uniform mesh.

4 Numerical Examples

In this section, we have applied our method on three linear singular perturbation problems with left-end boundary layer and two with right-end boundary layer. We have presented numerical results as well as maximum absolute errors of the examples. These examples have been chosen because they have been widely discussed in literature.

Example 1. Consider the following homogeneous singular perturbation problem

$$\varepsilon y''(x) + y'(x) - y(x) = 0\ ;\ \ x \in [0,\ 1]$$

with $y(0) = 1$ and $y(1) = 1$.

The exact solution is given by

$$y(x) = [(e^{m_2} - 1)e^{m_1 x} + (1 - e^{m_1})e^{m_2 x}]/[e^{m_2} - e^{m_1}]$$

where $m_1 = (-1 + \sqrt{1+4\varepsilon})/(2\varepsilon)$ and $m_2 = (-1 - \sqrt{1+4\varepsilon})/(2\varepsilon)$

The numerical results are given in Tables 1 and 2 for different values of ε .

Example 2. Now consider the following non-homogeneous singular perturbation problem

$$\varepsilon y''(x) + y'(x) = 1 + 2x\ ;\ \ x \in [0,1]$$

with $y(0) = 0$ and $y(1) = 1$.

Clearly this problem has a boundary layer at x = 0. The exact solution is given by

$$y(\mathrm{x}) = \frac{\left(e^{-x} - e^{-x/\varepsilon}\right)}{\left(e^{-1} - e^{-1/\varepsilon}\right)}$$

The numerical results are presented in Tables 3 and 4 for different values of ε.

Example 3. Consider the following singular perturbation problem

$$\varepsilon y''(x) + y'(x) = 2\,;\ \ x \in [0,1]$$

with $y(0) = 0$ and $y(1) = 1$.

The exact solution is given by $y(x) = 2x + \dfrac{1 - e^{-\left(x/\varepsilon\right)}}{e^{-\left(1/\varepsilon\right)} - 1}$.

The numerical results are presented in Tables 5 and 6 for different values of ε.

Example 4. Consider the following singular perturbation problem

$$\varepsilon y''(x) - y'(x) = 0\,;\ \ x \in [0,\ 1]$$

with $y\ (0) = 1$ and $y\ (1) = 0$.

Clearly, this problem has a boundary layer at x = 1 i.e., at the right end of the underlying interval.

The exact solution is given by $y(x) = \dfrac{\left(e^{(x-1)/\varepsilon} - 1\right)}{\left(e^{-1/\varepsilon} - 1\right)}$

The numerical results are presented in Tables 7 and 8 for different values of ε.

Example 5. Consider the following singular perturbation problem

$$\varepsilon y''(x) - y'(x) - (1+\varepsilon)y(x) = 0\,;\ \ x \in [0,\ 1]$$

with $y(0) = 1+\exp(-(1+\varepsilon)/\varepsilon)$; and $y(1) = 1+1/e$.

Clearly this problem has a boundary layer at x = 1. The exact solution is given by

$$y(x) = e^{(1+\varepsilon)(x-1)/\varepsilon} + e^{-x}$$

The numerical results are presented in Tables 9 and 10 for different values of ε.

We compare the maximum absolute errors of these examples with the proposed method and second order central finite difference scheme to show the advantage of the method. The maximum absolute errors of the examples are presented in Tables 11 and 12.

Table 1. Numerical results of example 1 with $h = 10^{-2}, \varepsilon = 10^{-4}$

x	Numerical solution	Exact solution
0.00	1.00000000	1.00000000
0.01	0.37795437	0.37161347
0.02	0.37543377	0.37534787
0.03	0.37914317	0.37911980
0.04	0.38295237	0.38292963
0.05	0.38680048	0.38677775
0.10	0.40662887	0.40660624
0.20	0.44938713	0.44936490
0.30	0.49664155	0.49662005
0.40	0.54886492	0.54884455
0.50	0.60657973	0.60656098
0.60	0.67036343	0.67034685
0.70	0.74085418	0.74084044
0.80	0.81875725	0.81874712
0.90	0.90485206	0.90484646
1.00	1.00000000	1.00000000

Maximum error = 6.3409e-003

Table 2. Numerical results of example 1 with $h = 10^{-2}, \varepsilon = 10^{-5}$

x	Numerical solution	Exact solution
0.00	1.00000000	1.00000000
0.01	0.37362461	0.37158036
0.02	0.37535308	0.37531477
0.03	0.37911862	0.37908671
0.04	0.38292844	0.38289656
0.05	0.38677656	0.38674469
0.10	0.40660505	0.40657331
0.20	0.44936373	0.44933255
0.30	0.49661892	0.49658877
0.40	0.54884348	0.54881492
0.50	0.60655999	0.60653369
0.60	0.67034598	0.67032272
0.70	0.74083971	0.74082044
0.80	0.81874659	0.81873239
0.90	0.90484617	0.90483832
1.00	1.00000000	1.00000000

Maximum error = 2.0442e-003

Table 3. Numerical results of example 2 with $h = 10^{-2}, \varepsilon = 10^{-4}$

x	Numerical solution	Exact solution
0.00	0.00000000	0.00000000
0.01	-0.97981098	-0.98970200
0.02	-0.97940951	-0.97940400
0.03	-0.96900981	-0.96890600
0.04	-0.95831213	-0.95820800
0.05	-0.94741346	-0.94731000
0.10	-0.88991988	-0.88982000
0.20	-0.75993184	-0.75984000
0.30	-0.60994274	-0.60986000
0.40	-0.43995270	-0.43988000
0.50	-0.24996186	-0.24990000
0.60	-0.03997033	-0.03991999
0.70	0.190021742	0.190060000
0.80	0.440014242	0.440040000
0.90	0.710007037	0.710020000
1.00	1.000000000	1.000000000

Maximum error = 9.8910e-003

Table 4. Numerical results of example 2 with $h = 10^{-2}, \varepsilon = 10^{-5}$

x	Numerical solution	Exact solution
0.00	0	0
0.01	-0.98676244	-0.98988020
0.02	-0.97963626	-0.97958040
0.03	-0.96914631	-0.96908060
0.04	-0.95844628	-0.95838080
0.05	-0.94754620	-0.94748100
0.10	-0.89004563	-0.88998200
0.20	-0.76004359	-0.75998400
0.30	-0.61004049	-0.60998600
0.40	-0.44003645	-0.43998800
0.50	-0.25003162	-0.24999000
0.60	-0.04002611	-0.03999199
0.70	0.189979936	0.190006000
0.80	0.439986392	0.440004000
0.90	0.709993124	0.710002000
1.00	1.000000000	1.000000000

Maximum error = 3.1178e-003

Table 5. Numerical results of example 3 with $h = 10^{-2}, \varepsilon = 10^{-4}$

x	Numerical solution	Exact solution
0.00	0.00000000	0.00000000
0.01	-0.97019349	-0.98000000
0.02	-0.96009095	-0.96000000
0.03	-0.94018736	-0.94000000
0.04	-0.92018579	-0.92000000
0.05	-0.90018326	-0.90000000
0.10	-0.80017073	-0.80000000
0.20	-0.60014663	-0.60000000
0.30	-0.40012381	-0.40000000
0.40	-0.20010228	-0.20000000
0.50	-8.20329e-05	4.44089e-16
0.60	0.199936937	0.200000000
0.70	0.399954625	0.400000000
0.80	0.599971032	0.600000000
0.90	0.799986156	0.800000000
1.00	1.000000000	1.000000000

Maximum error = 9.8065e-003

Table 6. Numerical results of example 3 with $h = 10^{-2}, \varepsilon = 10^{-5}$

x	Numerical solution	Exact solution
0.00	0	0
0.01	-0.97701164	-0.98000000
0.02	-0.96018336	-0.96000000
0.03	-0.94019085	-0.94000000
0.04	-0.92018828	-0.92000000
0.05	-0.90018570	-0.90000000
0.10	-0.80017296	-0.80000000
0.20	-0.60014847	-0.60000000
0.30	-0.40012530	-0.40000000
0.40	-0.20010345	-0.20000000
0.50	-8.29226e-05	4.44089e-16
0.60	0.199936295	0.200000000
0.70	0.399954196	0.400000000
0.80	0.599970781	0.600000000
0.90	0.799986049	0.800000000
1.00	1.000000000	1.000000000

Maximum error = 2.9884e-003

Table 7. Numerical results of example 4 with $h = 10^{-2}, \varepsilon = 10^{-4}$

x	**Numerical solution**	**Exact solution**
0	1	1
0.10	1.00000640	1
0.20	1.00001281	1
0.30	1.00001922	1
0.40	1.00002563	1
0.50	1.00003204	1
0.60	1.00003844	1
0.70	1.00004485	1
0.80	1.00005126	1
0.90	1.00005767	1
0.95	1.00006087	1
0.96	1.00006151	1
0.97	1.00006116	1
0.98	0.99996283	1
0.99	0.99006474	1
1	0	0

Maximum error = 9.9353e-003

Table 8. Numerical results of example 4 with $h = 10^{-2}, \varepsilon = 10^{-5}$

x	**Numerical solution**	**Exact solution**
0	1	1
0.10	1.00000658	1
0.20	1.00001316	1
0.30	1.00001975	1
0.40	1.00002633	1
0.50	1.00003291	1
0.60	1.00003950	1
0.70	1.00004608	1
0.80	1.00005266	1
0.90	1.00005925	1
0.95	1.00006254	1
0.96	1.00006320	1
0.97	1.00006382	1
0.98	1.00005438	1
0.99	0.99688111	1
1	0	0

Maximum error = 3.1189e-003

Table 9. Numerical results of example 5 with $h = 10^{-2}, \varepsilon = 10^{-4}$

x	Numerical solution	Exact solution
0	1	1
0.10	0.90484301	0.90483741
0.20	0.81874087	0.81873075
0.30	0.74083196	0.74081822
0.40	0.67033662	0.67032004
0.50	0.60654941	0.60653065
0.60	0.54883200	0.54881163
0.70	0.49660680	0.49658530
0.80	0.44935119	0.44932896
0.90	0.40659228	0.40656965
0.95	0.38676374	0.38674102
0.96	0.38291562	0.38289288
0.97	0.37910677	0.37908303
0.98	0.37543376	0.37531109
0.99	0.38159532	0.37157669
1	1.36787944	1.36787944

Maximum error = 1.0019e-002

Table 10. Numerical results of Example 5 with $h = 10^{-2}, \varepsilon = 10^{-5}$

x	Numerical solution	Exact solution
0	1	1
0.10	0.90484526	0.90483741
0.20	0.81874495	0.81873075
0.30	0.74083749	0.74081822
0.40	0.67034330	0.67032004
0.50	0.60655696	0.60653065
0.60	0.54884019	0.54881163
0.70	0.49661545	0.49658530
0.80	0.44936014	0.44932896
0.90	0.40660139	0.40656965
0.95	0.38677288	0.38674102
0.96	0.38292476	0.38289288
0.97	0.37911496	0.37908303
0.98	0.37535313	0.37531109
0.99	0.37479214	0.37157669
1	1.36787944	1.36787944

Maximum error = 3.2155e-003

Table 11. Maximum absolute errors of the examples by the proposed method

h	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Example 1				
$\varepsilon = 10^{-6}$	4.5865e-003	1.6473e-003	9.0044e-004	7.0621e-004
$\varepsilon = 10^{-7}$	4.0854e-003	1.1772e-003	4.4882e-004	2.6450e-004
$\varepsilon = 10^{-8}$	3.9267e-003	1.0284e-003	3.0579e-004	1.2461e-004
Example 2				
$\varepsilon = 10^{-6}$	1.4680e-002	3.8650e-003	9.8088e-004	7.7618e-004
$\varepsilon = 10^{-7}$	1.5668e-002	3.9500e-003	1.0230e-003	2.5626e-004
$\varepsilon = 10^{-8}$	1.5981e-002	4.0618e-003	1.0365e-003	2.6296e-004
Example 3				
$\varepsilon = 10^{-6}$	2.5365e-002	6.5616e-003	1.7900e-003	5.3799e-004
$\varepsilon = 10^{-7}$	2.6204e-002	6.8541e-003	1.7934e-003	4.6781e-004
$\varepsilon = 10^{-8}$	2.6469e-002	7.0894e-003	1.7942e-003	4.6810e-004
Example 4				
$\varepsilon = 10^{-6}$	8.0531e-003	2.2763e-003	6.0739e-004	8.5536e-004
$\varepsilon = 10^{-7}$	8.8667e-003	2.2836e-003	6.1002e-004	1.6114e-004
$\varepsilon = 10^{-8}$	9.1243e-003	2.3422e-003	6.1064e-004	1.5764e-004
Example 5				
$\varepsilon = 10^{-6}$	5.0063e-003	2.0391e-003	1.2797e-003	1.0795e-003
$\varepsilon = 10^{-7}$	4.2183e-003	1.3013e-003	5.6886e-004	3.8264e-004
$\varepsilon = 10^{-8}$	3.9687e-003	1.0676e-003	3.4376e-004	1.6198e-004

Table 12. Maximum absolute errors of the examples with central difference scheme

h	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Example 1				
$\varepsilon = 10^{-6}$	1.2221e+000	1.3394e+000	1.3992e+000	1.4271e+000
$\varepsilon = 10^{-7}$	1.2222e+000	1.3396e+000	1.4002e+000	1.4306e+000
$\varepsilon = 10^{-8}$	1.2222e+000	1.3397e+000	1.4003e+000	1.4310e+000
Example 2				
$\varepsilon = 10^{-6}$	7.8124e+003	1.9531e+003	4.8825e+002	1.2206e+002
$\varepsilon = 10^{-7}$	7.8125e+004	1.9531e+004	4.8828e+003	1.2207e+003
$\varepsilon = 10^{-8}$	7.8125e+005	1.9531e+005	4.8828e+004	1.2207e+004
Example 3				
$\varepsilon = 10^{-6}$	7.8124e+003	1.9531e+003	4.8825e+002	1.2206e+002
$\varepsilon = 10^{-7}$	7.8125e+004	1.9531e+004	4.8828e+003	1.2207e+003
$\varepsilon = 10^{-8}$	7.8125e+005	1.9531e+005	4.8828e+004	1.2207e+004
Example 4				
$\varepsilon = 10^{-6}$	7.8124e+003	1.9531e+003	4.8825e+002	1.2206e+002
$\varepsilon = 10^{-7}$	7.8125e+004	1.9531e+004	4.8828e+003	1.2207e+003
$\varepsilon = 10^{-8}$	7.8125e+005	1.9531e+005	4.8828e+004	1.2207e+004
Example 5				
$\varepsilon = 10^{-6}$	1.1979e+000	1.2526e+000	1.2808e+000	1.2914e+000
$\varepsilon = 10^{-7}$	1.1980e+000	1.2530e+000	1.2822e+000	1.2969e+000
$\varepsilon = 10^{-8}$	1.1981e+000	1.2530e+000	1.2824e+000	1.2975e+000

5 Discussions and Conclusions

We have discussed the numerical integration with exponential fitting factor for singularly perturbed two-point boundary value problems. Here, we replaced the given SPTPBVP by an asymptotically equivalent delay differential equation using deviating argument. Then, numerical integration with exponential fitting factor is employed to obtain a tridiagonal system which is solved efficiently by Thomas algorithm. We presented the convergence analysis of the proposed method and it is proved as first order convergence. Model examples are solved and the numerical results are compared with exact solution. We also presented the maximum absolute errors of the examples with the proposed method and second order central difference scheme. We observed that our method gives good results when $\varepsilon << h$. This method is conceptually simple, easy to use and is readily adaptable for computer implementation with a modest amount of problem preparation. Further, it is also observed that the accuracy predicted can always be achieved with a little computational effort.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Bellman R. Perturbation Techniques in Mathematics, Physics and Engineering. Holt: Rinehart: Winston: New York; 1964.

[2] Bender CM, Orszag SA. Advanced Mathematical Methods for Scientists and Engineers. McGraw-Hill: New York; 1978.

[3] Doolan EP, Miller JJH, Schilders WHA. Uniform Numerical Methods for problems with Initial and Boundary Layers. Boole Press: Dublin; 1980.

[4] El’sgol’ts LE, Norkin SB. Introduction to the Theory and Applications of Differential Equations with Deviating Arguments. Academic Press: New York; 1973.

[5] Kevorkian J, Cole JD. Perturbation Methods in Applied Mathematics, Springer-Verlag, New York; 1981.

[6] Miller JJH, O’Riordan E, Shishkin GI. Fitted Numerical Methods for Singular Perturbation Problems. World Scientific: River Edge: NJ; 1996.

[7] Nayfeh AH. Introduction to Perturbation Techniques. Wiley: New York; 1981.

[8] Nayfeh AH. Problems in Perturbation. Wiley: New York; 1985.

[9] O’Malley RE. Introduction to Singular Perturbations. Academic Press: New York; 1974.

[10] Pearson CE. On a differential equation of boundary layer type. J. Math Phy. 1968;47:134-154.

[11] Smith DR. Singular Perturbation Theory – An Introduction with applications. Cambridge University Press: Cambridge; 1985.

[12] Soujanya GBSL, Reddy YN, Phaneendra K. Numerical solution of singular perturbation problems via deviating argument and exponential fitting. American Journal of Computational and Applied Mathematics. 2012;2(2):49-54.

[13] Brian J. McCartin, Exponential fitting of the delayed recruitment/renewal equation, Journal of Computational and Applied Mathematics. 2001;136:343-356.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=225&id=6&aid=1502