NOVI SAD J. MATH.
VOL. 30, NO. 1, 2000, 155-164



# THE MESH CHOPPING ALGORITHM FOR SINGULARLY PERTURBED BOUNDARY VALUE PROBLEM

**Dragoslav Herceg[1], Helena Maličić[1]**

**Abstract.** In [3], a parallel mesh chopping algorithm was presented for solving a class of two-point boundary value problems. Here, we use this algorithm without parallelism for solving singularly perturbed boundary value problems. A Bakhvalov type of mesh discretization with second- and fourth-order difference schemes are used, and the corresponding discrete analogues are solved by Newton's method.

*AMS Mathematics Subject Classification (1991):* 65L10

*Key words and phrases:* singular perturbation, boundary value problem, Bakhvalov type mesh, finite differences, Newton's method

## 1. Introduction

Singularly perturbed differential equations represent mathematical models of the phenomena that are studied in many branches of applied mathematics. They have been treated numerically in numerous papers - let us just mention [4], where a survey was given. Due to a small perturbation parameter $\varepsilon > 0$ multiplying the highest-order derivative terms, the solution of the singularly perturbed problem has one or more interior or boundary layers. These layers represent narrow regions of the domain of the differential equation where the solution changes in an abrupt manner. Their presence causes severe computational difficulties for standard numerical methods, as well as in error analysis, since the aim is to obtain error estimates independent of $\varepsilon$.

Here we shall consider a nonlinear singularly perturbed boundary value problem

$$\begin{aligned} -\varepsilon^2 u'' + c(x,u) &= 0, \qquad x \in I = [0,1], \\ u(0) = u(1) &= 0, \end{aligned} \tag{1}$$

where $\varepsilon \in (0, \varepsilon_0)$, $\varepsilon_0 \ll 1$. Throughout we assume

$$c \in C^2(I \times \mathbf{R}), \qquad 0 < \gamma^2 \leq c_u(x,u), \qquad x \in I, \quad u \in \mathbf{R}, \tag{2}$$

[1] Institute of Mathematics, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia, e-mail: {hercegd, helena}@unsim.im.ns.ac.yu

which guarantees that the problem (1) has a unique and smooth solution $u_\varepsilon \in C^4[0,1]$, with (in general) two boundary layers at $x = 0$ and $x = 1$. Since the layers are of width $\mathcal{O}(\varepsilon)$, it is necessary to use a special discretization mesh which gives considerable amount of mesh points in these layers. We shall use a nonequidistant mesh of Bakhvalov type from [5], combined with two schemes for approximating $u''(x)$. The first one will be a standard second-order central difference scheme, [2], and the second a combination presented in [6] which involves a fourth order Hermite scheme. In both cases, a uniform convergence of the second- and fourth-order has been proved (the term uniform means uniform in $\varepsilon$).

The discrete analogue of problem (1) will be denoted as

$$F(w) = 0 \tag{3}$$

where $F : \mathbf{R}^{n+1} \to \mathbf{R}^{n+1}$ is a nonlinear mapping of a special form. The solution $w^*$ of (3) obtained by Newton's method is a numerical approximation to $u_\varepsilon$.

The authors in [3] developed a "parallel mesh chopping" algorithm for solving the two-point boundary value problem of the form

$$y' = f(x,y), \qquad y(0) = A, \qquad y(1) = B, \tag{4}$$

where $\frac{\partial f}{\partial y} \geq 0$ and $\frac{\partial f}{\partial y}$ is continuous on $[0,1] \times \mathbf{R}$. The basic idea is to split the $[0,1]$ interval into $p$ different divisions, where each of them consists of $N$ or $N+1$ unequal subintervals. Approximating the first derivative in (4), in each division, the corresponding discrete analogues are $(N+1) \times (N+1)$, or $N \times N$ nonlinear systems of equations, which are then solved on $p$ processors.

In our work, a "mesh chopping" algorithm will be used in mesh discretization for (1). As a result we have to solve $p$ nonlinear systems with dimensions smaller than $n$, which gives a considerable reduction in CPU time, as the numerical examples will demonstrate. A disadvantage of this approach is the loss of accuracy - it will be shown that, in some cases, preserving second-order accuracy requires the use of approximation of higher order for $u''$.

This paper is organized as follows. In the next section mesh discretization and approximations of the second derivative in (1) will be introduced, together with convergence theorems. In Section 3, the "mesh chopping" algorithm adapted to solving singularly perturbed problem will be considered. It will be presented in a somewhat different manner than in [3]. In the last section, results of numerical experiments show the efficiency of this method in reducing the CPU time for solving (1).

## 2. Mesh discretization and discrete analogue

One of the usual techniques for solving the problem (1) is forming special mesh discretization which will resolve boundary layers, and using a difference

scheme of a certain order, adjusted to the chosen mesh. We shall consider the mesh generating function given in [5] with

$$\lambda(t) = \begin{cases} \mu(t) := \dfrac{a\varepsilon t}{q-t}, & t \in [0, \alpha], \\ \mu(\alpha) + \mu'(\alpha)(t-\alpha), & t \in [\alpha, 0.5], \\ 1 - \lambda(1-t), & t \in [0.5, 1]. \end{cases} \tag{5}$$

Here $q \in (0, 0.5)$, $a \in (0, q/\varepsilon)$ and $\alpha$ is the abscissa of the contact point of the tangent line from $(0.5, 0.5)$ to $\mu(t)$:

$$\alpha = \frac{q - \sqrt{aq\varepsilon(1 - 2q + 2a\varepsilon)}}{1 + 2a\varepsilon}.$$

Usually, the set of mesh points is

$$I_h = \{x_i = \lambda(ih) : i = 0, 1, \ldots, n\}, \tag{6}$$

where $n \in \mathbf{N}$, $h = n^{-1}$.

As in [2], for approximation of $-\varepsilon^2 u''(x)$ we shall use the central difference scheme, so the corresponding discrete analogue $F(w) = 0$ is

$$\begin{aligned} &F_0 := w_0 = 0, \\ &F_i := \varepsilon^2 (a_1(i) w_{i-1} + a_0(i) w_i + a_2(i) w_{i+1}) + c_i = 0, \\ &\qquad i = 1, 2, \ldots, n-1, \\ &F_n := w_n = 0, \end{aligned} \tag{7}$$

where

$$a_1(i) = \frac{-2}{h_i(h_i + h_{i+1})}, \quad a_0(i) = \frac{2}{h_i h_{i+1}}, \quad a_2(i) = -a_1(i) - a_0(i), \tag{8}$$

and $h_i = x_i - x_{i-1}$, $c_i = c(x_i, w_i)$. The uniqueness of the solution $w^*$ of the problem (7) and the convergence of the Newton method for solving (7) are proven in [2]. The order of convergence for this scheme on the mesh (5) is presented in [6] in the following theorem.

**Theorem 2.1** *Suppose that the condition* (2) *is satisfied. Let* $u_\varepsilon$ *be the solution to the problem* (1), $u_{\varepsilon,h} = [u_\varepsilon(x_0), u_\varepsilon(x_1), \ldots, u_\varepsilon(x_n)]^\top$ *and* $w^*$ *the solution to the problem* (7). *Then*

$$\|u_{\varepsilon,h} - w^*\| \le Mh^2,$$

*where* $M > 0$ *is a constant independent of* $\varepsilon$ *and* $h$.

For approximation of $-\varepsilon^2 u''(x)$ in (1) we shall also use the fourth-order scheme from [6]. The corresponding discrete analogue is

$$
\begin{aligned}
&F_0 := w_0 = 0, \\
&F_i := \varepsilon^2 \left(a_1(i)\, w_{i-1} + a_0(i)\, w_i + a_2(i)\, w_{i+1}\right) \\
&\quad + b_1(i)\, c_{i-1} + b_0(i)\, c_i + b_2(i)\, c_{i+1} = 0, \\
&\quad i = 1, 2, \ldots, n-1, \\
&F_n := w_n = 0,
\end{aligned} \tag{9}
$$

with

$$
b_1(i) = -a_1(i)\,\frac{h_i^2 - h_{i+1}^2 + h_i h_{i+1}}{12}, \qquad b_2(i) = -a_2(i)\,\frac{h_{i+1}^2 - h_i^2 + h_i h_{i+1}}{12},
$$

$$
b_0(i) = a_0(i)\,\frac{h_i^2 + h_{i+1}^2 + 3h_i h_{i+1}}{12},
$$

where $a_1(i), a_0(i), a_2(i)$ are as in (8).

The fourth order of this scheme follows from the following theorem.

**Theorem 2.2.** *[6] Let $u_\varepsilon$ be the solution to the problem* (1),

$$
u_{\varepsilon,h} = \left[u_\varepsilon(x_0), u_\varepsilon(x_1), \ldots, u_\varepsilon(x_n)\right]^{\mathsf{T}}
$$

*and $w^*$ the solution to the problem* (9). *Then*

$$
\|u_{\varepsilon,h} - w^*\| \leq M h^4,
$$

*where $M > 0$ is a constant independent of $\varepsilon$ and $h$.*

## 3. "Mesh chopping" algorithm

In this section we shall describe the mesh chopping algorithm from [3] adjusted to the mesh discretization (6).

Let $N \in \mathbf{N}$, $p = 2^s$, $s \in \mathbf{N}$. The interval $[0,1]$ is divided into $p$ different divisions. Each division (marked as $i$) consists of the uniformly distributed points $\tau_{ij}$, $j = 1, 2, \ldots, N$, including $\tau_{i0} = 0$ and if $i \neq p$, also $\tau_{i,N+1} = 1$. These points can be represented as

$$
\tau_{ij} = \begin{cases} 0, & j = 0, \quad i = 1, 2, \ldots, p, \\ \dfrac{i}{Np} + \dfrac{j-1}{N}, & j = 1, 2, \ldots, N, \quad i = 1, 2, \ldots, p, \\ 1, & j = N+1, \quad i = 1, 2, \ldots, p-1. \end{cases}
$$

Let $t_k \in [0,1]$, $k = 0, 1, \ldots, n = Np$, be the points defined with

$$t_k = t_{pi+j} = \begin{cases} \tau_{i+1,j}, & j = 1, 2, \ldots, p-1, \\ \tau_{ip}, & j = 0, \end{cases} \qquad k = 0, 1, \ldots, n.$$

From this definition it is obvious that $t_k$ are uniformly distributed points with the step $n^{-1}$. The $p$ ($p = 4$) different divisions of the $[0,1]$ interval, with $N = 4$ and $h = 1/N$ are depicted in Figure 1.

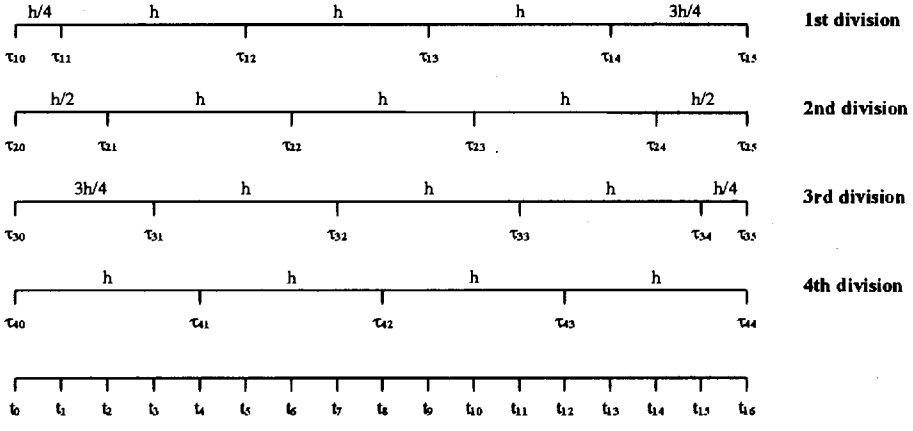


Figure 1

As in [3], the problem (1) is solved separately on each division, but, because of the impossibility of using parallel processors, we solved these problems one by one using one processor. We used the difference schemes (7) and (9), and with the points $\tau_{ij}$ we formed the discretization mesh as

$$\begin{aligned} I_{h,i} &= \{x_j = \lambda(\tau_{ij}) : j = 0, 1, \ldots, N+1\}, \quad i = 1, 2, \ldots, p-1, \\ I_{h,p} &= \{x_j = \lambda(\tau_{ij}) : j = 0, 1, \ldots, N\}, \qquad i = p. \end{aligned} \tag{10}$$

On each division, discrete analogues are the systems of dimension $(N+1) \times (N+1)$, for $i = p$, or $N \times N$, for $i \neq p$. These "smaller" systems are then solved using Newton's method and the obtained solutions are used for constructing the vector $w$ which represents a numerical approximation to $u_\varepsilon$. This means

that instead of solving one nonlinear system of dimension $n = Np$, we solved $p$ nonlinear systems of smaller dimension ($N$ or $N+1$). If with $w^i$, $i = 1, 2, \ldots, p$, we denote the solution of the "smaller" nonlinear system on the division $i$, then the vector $w$ is

$$w = [w_0, w_1, \ldots, w_n]^\mathsf{T}, \qquad n = Np,$$

where

$$w_k = w_{pi+j} = \begin{cases} w_{i+1}^j, & j = 1, 2, \ldots, p-1, \\ w_i^p, & j = 0, \end{cases} \qquad k = 0, 1, \ldots, n.$$

It is to be expected that the total CPU time for solving $p$ "smaller" nonlinear systems will be smaller than the CPU time for solving one "large" system, as the numerical experiments show. However, the disadvantage of this method is a loss of accuracy. By analyzing the error of the scheme (9) it can be easily shown that in case $p = N$, the order of the scheme reduces from four to two. In case of using the scheme (7), the order of convergence remains the same.

## 4. Numerical examples

We tested several numerical examples which are usually used in the literature. First four problems have unknown solution, whereas the fifth and sixth problems are artificially constructed. The previously described method was tested only for $p = N$.

For comparison, first we solved these problems using the second-order scheme (7) with (6) as the mesh discretization and Newton's method for solving the corresponding discrete analogue (this method is denoted with "$N-2$"). The mesh chopping algorithm consisted of using the mesh discretization (10) together with the schemes (7) and (9) both of order two, as well as using Newton's method for solving the corresponding discrete analogues (these methods are denoted with "$CN-2$" and "$CN-4$" respectively). In all cases the starting vector for Newton's method was

$$[u_0(x_0), u_0(x_1), \ldots, u_0(x_n)]^\mathsf{T},$$

where $u_0$ is the solution to the reduced problem $c(x, u) = 0$.

Let $u_n^N$, $u_n^{C2}$ and $u_n^{C4}$ be the approximations for $u_\varepsilon$ obtained using the "$N-2$", "$CN-2$" and "$CN-4$" methods respectively, with $n$ points in the mesh discretization. For the first four examples we measured

$$\left\|u_{8192}^N - u^N\right\|_\infty, \qquad \left\|u_{8192}^N - u^{C2}\right\|_\infty, \qquad \left\|u_{8192}^N - u^{C4}\right\|_\infty,$$

where $\|\cdot\|_\infty$ denotes the standard discrete maximum norm on $[0, 1]$. For the examples 5 and 6, where the exact solution $u_\varepsilon$ is known, we measured

$$\left\|u_\varepsilon - u^N\right\|_\infty, \qquad \left\|u_\varepsilon - u^{C2}\right\|_\infty, \qquad \left\|u_\varepsilon - u^{C4}\right\|_\infty.$$

In all cases we also measured the total CPU time for solving the problem (1) using the methods "$N-2$", "$CN-2$" and "$CN-4$". The CPU time for the mesh chopping algorithms is obtained as the sum of all CPU times for solving the same problem on each division. All experiments were performed using the package *Mathematica 3.0.*

The parameters used for the first four examples are $a=1, q=0.4, \varepsilon=10^{-12}$.

**Example 1**

$$-\varepsilon^2 u'' + \left(u^2+u-0.75\right)\left(u^2+u-3.75\right) = 0, \qquad u(0) = u(1) = 0,$$

$$u_0(x) = -1.5$$

| $n = p \times N$ | $N-2$ | $CN-2$ | $CN-4$ |
|---|---|---|---|
| $64 = 8 \times 8$ | 1.61785 (−3)<br>0.66 | 1.86541 (−3)<br>1.32 | 7.39714 (−5)<br>3.52 |
| $256 = 16 \times 16$ | 1.00869 (−4)<br>2.47 | 1.03008 (−4)<br>3.62 | 2.73617 (−7)<br>4.18 |
| $1024 = 32 \times 32$ | 6.2131 (−6)<br>17.9 | 6.29423 (−6)<br>10.55 | 5.39856 (−9)<br>11.7 |
| $4096 = 64 \times 64$ | 2.95859 (−7)<br>234.87 | 4.08656 (−7)<br>33.01 | 1.06341 (−10)<br>43.17 |

**Example 2**

$$-\varepsilon^2 u'' + \frac{u-1}{5-4u} = 0, \qquad u(0) = u(1) = 0,$$

$$u_0(x) = 1$$

| $n = p \times N$ | $N-2$ | $CN-2$ | $CN-4$ |
|---|---|---|---|
| $64 = 8 \times 8$ | 3.37887 (−3)<br>0.6 | 3.94145 (−3)<br>0.94 | 1.61235 (−4)<br>1.37 |
| $256 = 16 \times 16$ | 2.10454 (−4)<br>2.69 | 2.47106 (−4)<br>2.85 | 9.20749 (−7)<br>3.63 |
| $1024 = 32 \times 32$ | 1.29619 (−5)<br>21.36 | 1.3405 (−5)<br>9.83 | 1.30861 (−8)<br>12.25 |
| $4096 = 64 \times 64$ | 6.17226 (−7)<br>282.87 | 8.2769 (−7)<br>37.52 | 2.0235 (−10)<br>47.84 |

**Example 3**

$$-\varepsilon^2 u'' + u - u^2 - x - x^2 + 2xu = 0, \qquad u(0) = u(1) = 0,$$

$$u_0(x) = x$$

| $n = p \times N$ | $N-2$ | $CN-2$ | $CN-4$ |
|---|---|---|---|
| $64 = 8 \times 8$ | 1.29389 (−3)<br>0.6 | 1.57289 (−3)<br>0.93 | 1.16967 (−4)<br>3.19 |
| $256 = 16 \times 16$ | 8.08157 (−5)<br>2.42 | 8.39031 (−5)<br>2.81 | 2.56553 (−7)<br>4.01 |
| $1024 = 32 \times 32$ | 4.97701 (−6)<br>18.01 | 5.11255 (−6)<br>9.23 | 5.00732 (−9)<br>12.03 |
| $4096 = 64 \times 64$ | 2.37001 (−7)<br>244.03 | 3.18767 (−7)<br>34.16 | 7.89346 (−11)<br>44.43 |

**Example 4**

$$-\varepsilon^2 u'' - u^2 + 1 + x = 0, \qquad u(0) = u(1) = 0,$$

$$u_0(x) = -\sqrt{1+x}$$

| $n = p \times N$ | $N-2$ | $CN-2$ | $CN-4$ |
|---|---|---|---|
| $64 = 8 \times 8$ | 1.74073 (−3)<br>0.5 | 2.00673 (−3)<br>0.66 | 1.23819 (−4)<br>2.8 |
| $256 = 16 \times 16$ | 1.08615 (−4)<br>2.25 | 1.11713 (−4)<br>2.09 | 2.73278 (−7)<br>2.91 |
| $1024 = 32 \times 32$ | 6.68828 (−6)<br>17.8 | 6.86439 (−6)<br>7.42 | 6.61626 (−9)<br>10.05 |
| $4096 = 64 \times 64$ | 3.18492 (−7)<br>245.13 | 4.31243 (−7)<br>29.76 | 1.08371 (−10)<br>39.05 |

For Examples 5 and 6 the parameters are $a = 1$, $q = 0.4$, $\varepsilon = 10^{-8}$.

**Example 5**

$$-\varepsilon^2 u'' + 0.5 \sin\left(1 - \frac{\exp\left(-\frac{x}{\varepsilon}\right) + \exp\left(-\frac{1-x}{\varepsilon}\right)}{1 + \exp\left(-\frac{1}{\varepsilon}\right)}\right) - 0.5 \sin u + u - 1 = 0,$$

$$u(0) = u(1) = 0,$$

$$u_\varepsilon(x) = 1 - \frac{\exp\left(-\frac{x}{\varepsilon}\right) + \exp\left(-\frac{1-x}{\varepsilon}\right)}{1 + \exp\left(-\frac{1}{\varepsilon}\right)}$$

| $n = p \times N$ | $N-2$ | $CN-2$ | $CN-4$ |
|---|---|---|---|
| $64 = 8 \times 8$ | 2.21154 (−3)<br>0.5 | 2.84385 (−3)<br>1.53 | 2.23262 (−4)<br>2.25 |
| $256 = 16 \times 16$ | 1.37798 (−4)<br>2.03 | 1.46759 (−4)<br>5. | 4.67894 (−7)<br>6.2 |
| $1024 = 32 \times 32$ | 8.61337 (−6)<br>14.33 | 8.73082 (−6)<br>17.63 | 8.88467 (−9)<br>22.58 |
| $4096 = 64 \times 64$ | 5.38708 (−7)<br>189.5 | 5.40152 (−7)<br>66.68 | 1.33319 (−10)<br>58.69 |
| $16384 = 128 \times 128$ | 3.44007 (−8)<br>3286.69 | 3.36766 (−8)<br>266.28 | 2.076939 (−12)<br>340.76 |

**Example 6**

$$-\varepsilon^2 u'' + u = -\cos^2(\pi x) - 2\varepsilon^2\pi^2\cos(2\pi x), \qquad u(0) = u(1) = 0,$$

$$u_\varepsilon(x) = \frac{\exp\left(-\frac{x}{\varepsilon}\right)+\exp\left(-\frac{1-x}{\varepsilon}\right)}{1+\exp\left(-\frac{1}{\varepsilon}\right)} - \cos^2(\pi x)$$

| $n = p \times N$ | $N-2$ | $CN-2$ | $CN-4$ |
|---|---|---|---|
| $64 = 8 \times 8$ | 1.67869 (−3)<br>0.21 | 2.03199 (−3)<br>0.38 | 1.25715 (−4)<br>0.49 |
| $256 = 16 \times 16$ | 1.0467 (−4)<br>0.88 | 1.09949 (−4)<br>1.21 | 3.58364 (−7)<br>1.87 |
| $1024 = 32 \times 32$ | 6.54093 (−6)<br>5.11 | 6.61063 (−6)<br>4.56 | 6.56399 (−9)<br>6.86 |
| $4096 = 64 \times 64$ | 4.09275 (−7)<br>48.88 | 4.0985 (−7)<br>18.07 | 1.01053 (−10)<br>26.92 |
| $16384 = 128 \times 128$ | 2.63295 (−8)<br>1438.72 | 2.55673 (−8)<br>75.36 | 1.57699 (−12)<br>110.62 |

## References

[1] Herceg D., Uniform fourth order difference scheme for a singular perturbation problem, Numer. Mat., 56 (1990), 675-693.

[2] Herceg D., Krejić N., On a numerical method for discrete analogues of boundary value problems, Nonlinear Analysis, Theory, Methods and Applications, 30 (1997), 9-15.

[3] Katti C.P., Goel S., A parallel mesh chopping algorithm for a class of two-point boundary value problems, Computers Math. Applic., 35 (1998), 121-128.

[4] Roos H.-G., Stynes M., Tobiska L., Numerical methods for singularly perturbed differential equations: convection diffusion and flow problems, Springer-Verlag, Berlin, 1996.

[5] Vulanović R., Mesh construction for discretization of singularly perturbed boundary value problems, Ph.D. thesis, University of Novi Sad, 1986.

[6] Vulanović R., On numerical solution of semilinear singular perturbation problems by using the Hermite scheme, Univ. u Novom Sadu, Zb. Rad. Prirod.-Mat. Fak. Ser. Mat., 3 (1993), 363-379.

*Received by the editors June 5, 2000.*