Grid equidistribution for reaction-diffusion problems in one dimension *

Natalia Kopteva†, Niall Madden‡ and Martin Stynes§

Abstract

The numerical solution of a linear singularly-perturbed reaction-diffusion two-point boundary value problem is considered. The method used is adaptive movement of a fixed number of mesh points by monitor-function equidistribution. A partly heuristic argument based on truncation error analysis leads to several suitable monitor functions, but also shows that the standard arc-length monitor function is unsuitable for this problem. Numerical results are provided to demonstrate the effectiveness of our preferred monitor function.

1 Introduction

We consider the linear reaction-diffusion two-point boundary value problem

$$Lu = -\varepsilon^2 u''(x) + b(x)u(x) = f(x) \text{ for } x \in (0,1), \quad u(0) = u(1) = 0. \tag{1.1}$$

where $b, f \in C^4[0,1]$ and $0 < \beta < b(x) \leq \bar{b}$ on $[0,1]$.

When ε is small, the solution u of (1.1) typically exhibits boundary layers at both ends of the interval $[0,1]$. Thus the problem is singularly perturbed. As a consequence, standard numerical methods for (1.1) may yield inaccurate results. Special numerical methods for (1.1) whose convergence behaviour is uniform in the parameter ε have been discussed in the literature — see, e.g., [16, 17, 19]. While special methods give satisfactory numerical results for (1.1), nevertheless some care is needed in extending them to more general situations such as a system of reaction-diffusion two-point boundary value problems [13, 15]. For this reason there is great current interest in the use of *adaptive* numerical methods for singularly perturbed differential equations in general, as such methods are intrinsically more general in their nature. Thus the aim of the present paper is to discuss and analyse an adaptive numerical method for (1.1).

Our adaptive method uses grid equidistribution of a monitor function combined with a standard difference scheme. Grid equidistribution has already been applied to convection-diffusion problems by several researchers (e.g., [3, 9, 12, 14]), where the arc-length of the computed solution and other monitor functions are considered. While equidistribution of the arc-length is intuitively a reasonable way of controlling an adaptive algorithm, nevertheless for reaction-diffusion problems such as (1.1)

*Research supported by the Boole Centre for Research in Informatics, National University of Ireland, Cork, Ireland

†Department of Mathematics and Statistics, University of Limerick, Limerick, Ireland; natalia.kopteva@ul.ie
This paper was written while the first author was visiting the Department of Mathematics, National University of Ireland, Cork, Ireland

‡Department of Mathematics, National University of Ireland, Galway, Ireland; niall.madden@nuigalway.ie

§Department of Mathematics, National University of Ireland, Cork, Ireland; m.stynes@ucc.ie

it may yield a solution that is much less accurate than that generated by the equidistribution of an alternative measure of the computed solution; compare Tables 5.1 and 5.2 below. Our new analysis throws light on why the arc-length monitor function is unsuitable for (1.1) — see Remark 4.3 — and also enables us to design new monitor functions that yield accurate solutions of (1.1). This illustrates the fact that *the form of the monitor function and the order of discrete derivatives used should depend on the order of convergence expected from the adaptive numerical method and is largely independent of the type of singularly perturbed problem (convection-diffusion or reaction-diffusion).* This observation has not been clearly elucidated in previously published papers, although for example the connection between the order of discrete derivative and the order of convergence was recognized implicitly in the first-order and second-order a posteriori analysis of convection-diffusion problems of [8]. Thus the arc-length monitor function is suitable for first-order convergent methods (as seen in [9]) but will not yield second-order convergence.

The structure of our paper is as follows: some facts about the solution u of (1.1) and stability bounds for the computed solution u^N (on an arbitrary mesh) are listed in §2, then in §3 a truncation error analysis yields bounds on the error $u-u^N$ in terms of the mesh spacing. These bounds are used in §4 to provide strong heuristic evidence for the choice of monitor function to be equidistributed in an adaptive algorithm. This section also shows that the arc-length monitor function is an unsatisfactory choice for (1.1). Numerical results using an adaptive algorithm are given in §5 to show that the rate of convergence predicted by our analysis holds true in practice for our preferred monitor function. Furthermore, this algorithm can easily be applied to semilinear analogues of (1.1); see §5.3 for an example.

Notation. Throughout the paper, C denotes a generic positive constant that is independent of ε and of the mesh. It may take different values in different places. When C is subscripted, it is a fixed constant that is independent of ε and of the mesh. The notation K is used to denote a generic quantity that depends on the mesh but is bounded by a fixed constant for all meshes. It may take different values in different places.

2 Preliminary results

The differential operator L satisfies a maximum principle since $b(\cdot) > 0$. Hence (1.1) has a unique solution u. Furthermore, an easy extension of [16, Chapter 6, Lemma 1] gives the following bounds for the derivatives of u.

Lemma 2.1. *Let i be a positive integer. Let $f(x), b(x) \in C^4[0,1]$ and let u be the solution of (1.1). Then for $0 \le k \le 4$ and for all $x \in [0,1]$,*

$$|u^{(k)}(x)| \le C[1 + \varepsilon^{-k} e(x,\beta)]$$

where $e(x,\beta) = e^{-\sqrt{\beta}x/\varepsilon} + e^{-\sqrt{\beta}(1-x)/\varepsilon}$.

For our numerical method we consider arbitrary meshes $\{x_0, x_1, \ldots, x_N\}$ with $0 = x_0 < x_1 < \cdots < x_N = 1$, where N, the discretization parameter, is a positive integer. For $i = 1, \ldots, N$, let $h_i = x_i - x_{i-1}$ be the local mesh size and set $\hbar_i = (h_i + h_{i+1})/2$ for each i.

Given a mesh function $v = \{v_i\}_0^N$, define the discrete difference operators

$$D^- v_i = (v_i - v_{i-1})/h_i, \quad Dv_i = (v_{i+1} - v_i)/\hbar_i \quad \text{and } \delta^2 v_i = DD^- v_i.$$

The computed solution $u^N = \{u_i^N\}_{i=0}^N$ is required to satisfy the standard finite difference discretization of (1.1):

$$L^N u_i^N := -\varepsilon^2 \delta^2 u_i^N + b_i u_i^N = f_i \text{ for } i = 1, \ldots, N-1, \qquad u_0^N = u_N^N = 0, \tag{2.1}$$

where $b_i = b(x_i)$ and $f_i = f(x_i)$. Then L^N satisfies a discrete maximum principle and it follows that (2.1) has a unique solution u^N.

For mesh functions $\{v_i\}$, the following mesh-dependent norms will be used:

$$\|v\|_\infty := \max_i |v_i|, \qquad |v|_{1,\infty} := \max_i |D^- v_i|, \qquad \|v\|_{1,\infty;\varepsilon^2} := \varepsilon^2 |v|_{1,\infty} + \|v\|_{-1,\infty},$$
$$\|v\|_{-1,\infty} := \min_{C \in \mathcal{R}} \max_{i=1,\ldots,N-1} \Big| \sum_{j=i}^{N-1} v_j \hbar_j - C \Big|.$$

In fact it can be shown [1] that

$$\|v\|_{-1,\infty} = \frac{1}{2} \Big(\min_i \Big| \sum_{j=i}^{N-1} v_j \hbar_j \Big| + \max_i \Big| \sum_{j=i}^{N-1} v_j \hbar_j \Big| \Big). \tag{2.2}$$

We now list some stability properties of the discretization (2.1). First, the discrete maximum principle implies the standard stability result

$$\|u^N\|_\infty \le \|f\|_\infty / \beta. \tag{2.3}$$

Recall that $b \in C^4[0,1]$. This implies that $\sum_{i=1}^N h_i |D^- b_i| \le C_1$ and $\sum_{i=1}^N h_i |D^-(1/b_i)| \le C_2$ for some constants C_1 and C_2. The second inequality in the next bound is a stability result, whose sharpness is shown by the first inequality.

Lemma 2.2. *[1, Theorem 3.1] The solution u^N of (2.1) satisfies*

$$C_3 \|f\|_{-1,\infty} \le \|u^N\|_{1,\infty;\varepsilon^2} \le C_4 \|f\|_{-1,\infty},$$

where $C_3 = (\bar{b} + C_1 + 2)^{-1}$, $C_4 = 3C_2 + 5/\beta$.

The next Lemma will enable us to interpolate between the bounds on $|u^N|_{1,\infty}$ and $\|u^N\|_{-1,\infty}$ that are implicit in Lemma 2.2.

Lemma 2.3. *[1, Lemma 2.4] Let $\{v_i\}$ be an arbitrary mesh function such that $v_0 = v_N = 0$. Let $\mu > 0$ be fixed. Then*

$$\|v\|_\infty \le 2\big(\mu |v|_{1,\infty} + \mu^{-1} \|v\|_{-1,\infty}\big)$$

Taking $\mu = \varepsilon$ in Lemma 2.3 and combining this inequality with Lemma 2.2, we get

Theorem 2.1. *[1, Theorem 3.2] The computed solution satisfies the stability inequality*

$$\|u^N\|_\infty \le 2 C_4 \varepsilon^{-1} \|f\|_{-1,\infty}.$$

3 Truncation error

In this section we begin by imitating the incisive truncation error analysis that Tihonov and Samarskii [24] (see also [11]) used to prove the supraconvergence of central difference schemes on nonuniform meshes. This is combined with the stability results of §2 to produce in Corollary 3.1 an error estimate suitable for our needs.

Let $e_i = u_i^N - u(x_i)$, for $i = 0, \ldots, N$, denote the error at x_i in the computed solution. Then

$$L^N e_i = r_i \quad \text{for } i = 1, \ldots, N-1, \qquad e_0 = e_N = 0,$$

where $r_i := f_i - L^N u_i$ is the truncation error at x_i. Set $x_{i-1/2} = (x_{i-1} + x_i)/2$ for each i.

Lemma 3.1. *For $i = 1, \ldots, N-1$,*

$$r_i = (\varepsilon/6)D\big(\varepsilon h_i^2 u'''(\eta_i)\big) + K\varepsilon^2 \hbar_i^2 u^{(4)}(\xi_i), \tag{3.1}$$

where $\eta_i \in (x_{i-1}, x_i)$ and $\xi_i \in (x_{i-1/2}, x_{i+1/2})$.

Proof. For each i,

$$r_i = (Lu)_i - L^N u_i = \varepsilon^2(\delta^2 u_i - u_i'') = \varepsilon^2 D(D^- u_i - u'_{i-1/2}) + \varepsilon^2(Du'_{i-1/2} - u_i''), \tag{3.2}$$

where $u_i'' = u''(x_i)$ and $u'_{i-1/2} = u'(x_{i-1/2})$, while $Du'_{i-1/2} = (u'_{i+1/2} - u'_{i-1/2})/\hbar_i$. For the first term here we have

$$D^- u_i - u'_{i-1/2} = (h_i^2/24)u'''(\hat\eta_i), \tag{3.3}$$

where $\hat\eta_i \in (x_{i-1}, x_i)$. Introducing the notation $w := u'$ and using Taylor series expansions, we can rewrite the second term in (3.2) as

$$Du'_{i-1/2} - u_i'' = Dw_{i-1/2} - w_i' = (h_{i+1} - h_i)w_i''/4 + K\hbar_i^2 w'''(\hat\xi_i),$$

where $\hat\xi_i \in (x_{i-1/2}, x_{i+1/2})$. But $h_{i+1} - h_i = (h_{i+1}^2 - h_i^2)/(2\hbar_i)$ and

$$w_i'' = w''_{i+1/2} - (h_{i+1}/2)w'''(\xi_i^+), \quad w_i'' = w''_{i-1/2} + (h_i/2)w'''(\xi_i^-),$$

where $\xi_i^\pm \in (x_{i-1/2}, x_{i+1/2})$. Hence for some $\xi_i \in (x_{i-1/2}, x_{i+1/2})$,

$$\begin{aligned} Du'_{i-1/2} - u_i'' &= (h_{i+1}^2 w''_{i+1/2} - h_i^2 w''_{i-1/2})/(8\hbar_i) + K\hbar_i^2 w'''(\xi_i) \\ &= (1/8)D(h_i^2 w''_{i-1/2}) + K\hbar_i^2 w'''(\xi_i) \\ &= (1/8)D(h_i^2 u'''_{i-1/2}) + K\hbar_i^2 u^{(4)}(\xi_i). \end{aligned}$$

Combining this with (3.2) and (3.3), for some $\eta_i \in (x_{i-1}, x_i)$ we get

$$\begin{aligned} r_i &= \varepsilon^2 D\big(h_i^2 u'''(\hat\eta_i)/24\big) + \varepsilon^2 D(h_i^2 u'''_{i-1/2}/8) + K\varepsilon^2\hbar_i^2 u^{(4)}(\xi_i) \\ &= (\varepsilon/6)D\big(\varepsilon h_i^2 u'''(\eta_i)\big) + K\varepsilon^2\hbar_i^2 u^{(4)}(\xi_i). \end{aligned}$$

This completes the proof. □

Theorem 3.1. *There exists a constant C such that*

$$\|e\|_\infty \le C\Big\{ \max_i \max_{x\in[x_{i-1},x_i]} \varepsilon h_i^2 |u'''(x)| + \max_i \max_{x\in[x_{i-1/2},x_{i+1/2}]} \hbar_i^2 \big[1 + |(bu)''(x)|\big] \Big\}. \tag{3.4}$$

Proof. The discrete solution u^N can be written as a sum of two mesh functions, where each corresponds to one of the terms in (3.1). Applying Theorem 2.1 to the first and (2.3) to the second of these functions, we see that

$$\|e\|_\infty \le C \left\{ \max_i \max_{x\in[x_{i-1},x_i]} \varepsilon h_i^2 |u'''(x)| + \max_i \max_{x\in[x_{i-1/2},x_{i+1/2}]} \varepsilon^2 \hbar_i^2 |u^{(4)}(x)| \right\}.$$

Differentiating (1.1) yields $\varepsilon^2 u^{(4)} = (bu)'' - f''$, and the desired result follows. □

In fact one can rewrite (3.4) in terms of lower-order derivatives:

Corollary 3.1. *There exists a constant C such that*

$$\|e\|_\infty \le C\Big\{ \max_i \max_{x\in[x_{i-1},x_i]} h_i^2 \big[1+|u''(x)|\big] + \max_i \max_{x\in[x_{i-1/2},x_{i+1/2}]} \hbar_i^2 \big[1+|(bu)''(x)|\big] \Big\}. \tag{3.5}$$

Proof. Set $b_0 = \sqrt{b(0)}$ and $b_1 = \sqrt{b(1)}$. A standard asymptotic analysis shows that the solution $u(x)$ of (1.1) can be written as

$$\begin{aligned} u(x) \;=\; & \frac{f(x)}{b(x)} - \frac{f(0)}{b(0)} e^{-b_0 x/\varepsilon} + \frac{f(0)b'(0)}{4b^2(0)} \left(\frac{b_0 x}{\varepsilon} + x \right) e^{-b_0 x/\varepsilon} \\ & - \frac{f(1)}{b(1)} e^{-b_1(1-x)/\varepsilon} + \frac{f(1)b'(1)}{4b^2(1)} \left(\frac{b_1(1-x)}{\varepsilon} + 1 - x \right) e^{-b_1(1-x)/\varepsilon} + \varepsilon^2 R(x), \end{aligned} \tag{3.6}$$

where $R(0)$ and $R(1)$ are exponentially small and

$$LR(x) = g(x,\varepsilon), \tag{3.7}$$

with

$$\begin{aligned} g(x,\varepsilon) \;=\; & -\left(\frac{f}{b}\right)''(x) + \left(\frac{x}{\varepsilon}\right)^2 \left[-\frac{b''(\eta_1)}{2} + \frac{f(0)b'(0)b'(\eta_2)}{4b^2(0)} \left(\frac{b_0 x}{\varepsilon} + 1 \right) \right] e^{-b_0 x/\varepsilon} \\ & + \left(\frac{1-x}{\varepsilon}\right)^2 \left[-\frac{b''(\eta_3)}{2} + \frac{f(1)b'(1)b'(\eta_4)}{4b^2(1)} \left(\frac{b_1(1-x)}{\varepsilon} + 1 \right) \right] e^{-b_1(1-x)/\varepsilon}, \end{aligned}$$

and $\eta_1(x), \eta_2(x) \in (0,x)$ while $\eta_3(x), \eta_4(x) \in (1-x,1)$. These assertions can be verified by applying L to the right-hand side of (3.6) and writing down Taylor expansions of $b(x) - b(0)$.

It is easy to see that $|g(x,\varepsilon)| \le C$ for $0 \le x \le 1$ and $0 < \varepsilon \le 1$. By [17, Lemmas 2.2 and 2.3],

$$|R^{(i)}(x)| \le C\varepsilon^{-i} \quad \text{for } 0 \le x \le 1 \text{ and } i = 0, 1.$$

It then follows from (3.7) and differentiating (3.7) that

$$|R^{(i)}(x)| \le C\varepsilon^{-i} \quad \text{for } 0 \le x \le 1 \text{ and } i = 2, 3. \tag{3.8}$$

We claim that for some constant C,

$$\varepsilon |u'''(x)| \le C(1 + |u''(x)|) \quad \text{for } 0 \le x \le 1. \tag{3.9}$$

Suppose that $0 \le x \le 1/2$; the case $1/2 \le x \le 1$ is similar. If $f(0) = 0$, then differentiating (3.6) three times and invoking (3.8), we have $\varepsilon|u'''(x)| \le C$. Thus suppose that $f(0) \ne 0$. Differentiating (3.6) twice, one sees that there exists a constant $x^* \in (0, 1/2]$ such that $|u''(x)| \ge C'\varepsilon^{-2} e^{-b_0 x/\varepsilon}$ for

$0 \le x \le x^*$ and some positive constant C'. Differentiating (3.6) three times and invoking (3.8), since x^* is fixed it follows that

$$\varepsilon|u'''(x)| \le \begin{cases} C(1+\varepsilon^{-2}e^{-b_0 x/\varepsilon}) & \text{if } 0 \le x \le x^*, \\ C & \text{if } x^* \le x \le 1/2. \end{cases}$$

Hence inequality (3.9) holds true.

This inequality, $\max\{h_i^2, h_{i+1}^2\} \le 4\hbar_i^2$, and Theorem 3.1 yield (3.5). □

The proof of (3.9) can easily be modified to yield

Corollary 3.2. *There exists a constant C such that*

$$\varepsilon^{1/2}|u'''(x)|^{1/2} \le C\left[1+|u'''(x)|^{1/3}\right] \quad \text{for } 0 \le x \le 1.$$

4 Monitor functions

Monitor functions are used by many authors (see, e.g., [4, 5, 6, 7, 9, 12, 18, 19]) to drive adaptive algorithms that produce layer-resolving meshes in solving differential equations. We first give a general description of this methodology, then use the theoretical results of §3 to choose monitor functions that are appropriate for (1.1).

A monitor function $M(x)$ is an arbitrary non-negative function defined on Ω. A mesh $\{x_i\}$ is said to equidistribute $M(\cdot)$ if

$$\int_{x_{i-1}}^{x_i} M(x)\,dx = \frac{1}{N}\int_0^1 M(x)\,dx \quad \text{for } i = 1, \ldots, N. \tag{4.1}$$

Adaptive algorithms based on M aim to construct a mesh that equidistributes M. This is done by computing a solution $\{u_i^N\}$ on the current mesh, forming an approximation (e.g., a piecewise constant function) $\tilde{M}(\cdot)$ of $M(\cdot)$ on this mesh, then choosing new mesh points that equidistribute $\tilde{M}$ on [0,1].

Thus the *equidistribution problem* is to find $\{(x_i, u_i^N)\}$, with the $\{u_i^N\}$ computed from the $\{x_i\}$ by means of (2.1), such that

$$h_i\tilde{M}_i = \frac{1}{N}\sum_{j=1}^{N} h_j\tilde{M}_j \quad \text{for } i = 1, 2, \ldots, N. \tag{4.2}$$

It is significant here that both the $\{x_i\}$ and $\{u_i^N\}$ are a priori unknown. Consequently, even though (1.1) is linear, the equidistribution problem requires the simultaneous solution of (2.1) and (4.2) and so is nonlinear.

For fixed N, the accuracy of a solution computed on an equidistributing mesh depends strongly on the choice of $M(\cdot)$. When an upwinded finite difference scheme is used to solve a convection-diffusion analogue of (1.1), one should use the arc-length monitor function $M(x) = \sqrt{1+(u'(x))^2}$, as is shown in [9], which is based on the a posteriori bound of [8, Theorem 3.2]. Here the choice of monitor function is guided by the order of accuracy of the numerical method — upwinding is first-order — and not by the type of singular perturbation problem (convection-diffusion). As solutions of reaction-diffusion problem have layers not dissimilar to those encountered in convection-diffusion problems, one might expect that the same arc-length monitor function would be suitable for (1.1),

but this will yield at best a first-order method. We shall use a difference scheme that is known to be (almost) second-order accurate on e.g. Shishkin/Bakhvalov meshes, and to reap the full benefit of this scheme one must use a fundamentally different monitor function that is based on discrete second-order derivatives of the computed solution.

A rigorous theoretical proof that a given monitor function will yield a satisfactory numerical method for (1.1) is extremely complicated — see the analysis in [9] for the convection-diffusion case. Thus we confine ourselves here to a more heuristic argument based on the a priori bounds of Theorem 3.1 and Corollary 3.1. (A posteriori bounds such as those of [8] are needed for a rigorous argument.) To be precise, we make the reasonable *assumption* that the following discrete analogues of (3.4) and (3.5) hold true and give sharp bounds on the error in the computed solution:

$$\|e\|_\infty \;\le\; C \max_i \left[\hbar_i^2 \big(1 + |\delta^2(b_i u_i^N)|\big) + h_i^2 \varepsilon |\delta^3 u_i^N|\right] \tag{4.3}$$

and

$$\|e\|_\infty \;\le\; C \max_i \hbar_i^2 \left[1 + \frac{|\delta^2 u_i^N| + |\delta^2(b_i u_i^N)|}{2}\right], \tag{4.4}$$

where $\delta^3 u_i^N = D^- \delta^2 u_i^N$. In theory one could of course replace the denominator 2 in (4.4) by 1.

Thus to make the error $\|e\|_\infty$ as small as possible, one should aim to construct a mesh that minimizes the right-hand side of (4.3) or (4.4). Let's consider (4.4) first. On our optimal mesh the sum

$$\hbar_i^2 \left[1 + \frac{|\delta^2 u_i^N| + |\delta^2(b_i u_i^N)|}{2}\right] \tag{4.5}$$

will be approximately independent of i, as otherwise one would adjust the mesh to decrease the right-hand side of (4.4). (This resembles the construction of Bakhvalov's mesh [2], where the truncation error is approximately the same at all nodes.) One may regard this as requiring

$$h_i^2 \left[1 + \frac{|\delta^2 u_{i-1}^N|^{1/2} + |\delta^2(b_{i-1} u_{i-1}^N)|^{1/2} + |\delta^2 u_i^N|^{1/2} + |\delta^2(b_i u_i^N)|^{1/2}}{4}\right]^2 \tag{4.6}$$

to be approximately independent of i.

To achieve (4.6) in the context of monitor functions, for each i associate with x_i the value

$$M_i = 1 + \frac{|\delta^2 u_i^N|^{1/2} + |\delta^2 (b u^N)_i|^{1/2}}{2\,\lambda}, \qquad \text{where} \quad \lambda := \max_i |u_i^N|^{1/2}. \tag{4.7}$$

The magnitude of u^N should not affect the suitability of the monitor function, so it is natural to normalize by λ. Define the piecewise constant $\tilde{M}_i = (M_{i-1} + M_i)/2$ on the interval $[x_{i-1}, x_i]$. Thus on an equidistributing mesh, for all i and j we have $h_i \tilde{M}_i = h_j \tilde{M}_j$, so $h_i^2 \tilde{M}_i^2 = h_j^2 \tilde{M}_j^2$, which agrees with the conclusion above that the sum in (4.5) should be essentially independent of i. The scaling of (4.7), where a square root is taken compared with (4.4), is chosen so that the first term there is independent of the mesh. Furthermore, with this scaling, $\sum_{j=1}^N h_j \tilde{M}_j \approx \int_0^1 M(x)\,dx$ neither blows up nor approaches zero when $\varepsilon \to 0$ (Lemma 2.1 shows that the layer component of $\int_0^1 |u''(x)|^r\,dx$ has this property only when $r = 1/2$).

Remark 4.1. *Many equivalent monitor functions M_i follow from (4.4), and in (4.7) we have chosen the most easily implemented one. In particular the denominator 2 could be replaced by any positive constant of moderate size.*

Similarly, (4.3) leads to the choice of monitor function

$$M_i = 1 + \frac{\varepsilon^{1/2}|\delta^3 u_i^N|^{1/2} + |\delta^2 (bu^N)_i|^{1/2}}{\lambda}. \tag{4.8}$$

The monitor function (4.8) corresponds to $M(x) = 1+\varepsilon^{1/2}|u'''(x)|^{1/2}+|u''(x)|^{1/2}$. Corollary 3.2 states that $\varepsilon^{1/2}|u'''(x)|^{1/2} \le C\left[1 + |u'''(x)|^{1/3}\right]$, and Lemma 2.1 yields $|u'''(x)| \le C[1 + \varepsilon^{-3}e(x,\beta)]$. Thus the explicit reference to $\varepsilon^{1/2}$ can be removed, while maintaining the scaling requirement that the corresponding term in $\int_0^1 M(x)\,dx$ tends neither to zero nor infinity as $\varepsilon \to 0$, by replacing $\varepsilon^{1/2}|u'''(x)|^{1/2}$ by $|u'''(x)|^{1/3}$. That is, instead of (4.8), one can try

$$M_i = 1 + \frac{|\delta^3 u_i^N|^{1/3} + |\delta^2 (bu^N)_i|^{1/2}}{\lambda}. \tag{4.9}$$

The presence of the constant 1 in all these monitor functions, which follows naturally from our truncation error analysis, prevents "mesh starvation" (i.e., too coarse a mesh) on any subinterval of [0,1] where $u'' \approx 0$.

Remark 4.2. *Beckett and Mackenzie [3, 4] consider the discrete monitor function*

$$M_i = \alpha(m) + |\delta^2 w_i|^{1/m}, \tag{4.10}$$

where $\alpha(m) = O(\varepsilon^{(m-2)/m})$, $w_i = u_i - f(x_i)/b(x_i)$ and $m \ge 2$ is a user-chosen parameter. When $m = 2$ this monitor function resembles (4.7), but the motivation for the choice of (4.10) in [3, 4] is unclear.

Remark 4.3. *(The arc-length monitor function) For constant $b(\cdot)$, one can express $\|e\|_\infty$ in terms of first-order derivatives by imitating the argument of Corollary 3.1 and invoking the inequality $\varepsilon|u''(x)| \le C(1 + |u'(x)|)$, obtaining*

$$\|e\|_\infty \le C \max_i \max_{x\in[x_{i-1},x_{i+1}]} \hbar_i^2 \left[1 + \varepsilon^{-1}|u'(x)|\right]. \tag{4.11}$$

To justify the use of the arc-length monitor function one would need an inequality of the form

$$\|e\|_\infty \le C \max_i \max_{x\in[x_{i-1},x_{i+1}]} \hbar_i^2 \left[1 + |u'(x)|^2\right]. \tag{4.12}$$

But (4.11) is weaker than (4.12): near $x = 0$ we expect that $|u'(x)| \approx C\varepsilon^{-1}e^{-\sqrt{b(0)}x/\varepsilon}$, so $|u'(x)|^2 \approx C^2\varepsilon^{-2}e^{-2\sqrt{b(0)}x/\varepsilon}$, which decays more rapidly than $\varepsilon^{-1}|u'(x)|$. Thus one would not expect the arc-length monitor function to yield satisfactory numerical results.

5 Numerical results

In §5.1 we descibe a grid generation method based on the equidistribution of the monitor function (4.7). Numerical experiments are presented in §5.2 for a linear problem; we demonstrate that the arc-length monitor function is inappropriate for this problem since more accurate and robust results can be obtained using the monitor function (4.7). Information about the behaviour of the iterative procedure and the distribution of the generated meshes is included. Finally, in §5.3 we give numerical results for a semilinear problem.

Numerical results for (4.8) and (4.9) are similar to those for (4.7) and are not reproduced here.

5.1 Mesh generation algorithm

In order to compute the equidistributed mesh and corresponding solution, a variant of the algorithm of [9, § 2] is used. For the monitor function (4.7) we proceed as follows:

1. Let $\{x^{(0)}\}_{i=0}^{N}$ be an initial uniform mesh with N intervals. Choose a constant $\gamma^\star > 1$ that controls when the algorithm terminates.

2. For a given mesh $\{x^{(k)}\}$ and computed solution $u^{(k)}(x)$, set

$$\tilde{M}_i^{(k)} = \frac{M_{i-1}^{(k)} + M_i^{(k)}}{2} \quad \text{for } i = 1, \dots N, \tag{5.1}$$

   where $M_i^{(k)}$ is the value of the monitor function computed at the ith interior node of the current mesh, and set $M_0^{(k)} = M_1^{(k)}$ and $M_N^{(k)} = M_{N-1}^{(k)}$.

3. Set $h_i^{(k)} = x_i^{(k)} - x_{i-1}^{(k)}$ for each i and set $L_0 = 0$ and $L_i = \sum_{j=1}^{i} h_j^{(k)} \tilde{M}_j^{(k)}$ for $i = 1, \dots, N$. Define

$$\gamma^{(k)} := \frac{N}{L_N} \max_{i=0,\dots,N} h_i^{(k)} \tilde{M}_i^{(k)}.$$

4. Set $Y_i = iL_N/N$ for $i = 0, \dots, N$. Interpolate (see Remark 5.1) to the points (L_i, x_i). Generate the new mesh $\{x^{(k+1)}\}$ by evaluating this interpolant at the Y_i for $i = 0, \dots, N$.

5. If $\gamma^{(k)} \le \gamma^\star$, then take $\{x^{(k+1)}\}$ as the final mesh and compute $u^{(k+1)}(x)$. Otherwise return to Step 2.

Remark 5.1. *In [9] piecewise linears are used for the interpolation in Step 4 of the algorithm. The numerical experiments below use instead piecewise cubic Hermite interpolation; this yields meshes that are more smoothly graded and the algorithm then takes fewer iterations to converge.*

A key observation of [9] is the realization that it is unnecessary to solve (4.2) exactly; instead, it is sufficient that on the final mesh one has

$$h_i \tilde{M}_i \le \frac{2}{N} \sum_{j=1}^{N} h_j \tilde{M}_j \quad \text{for } i = 1, 2, \dots, N.$$

This is equivalent to taking $\gamma^\star = 2$ in our algorithm. Except where we state otherwise, we shall take $\gamma^\star = 2$ throughout Section 5. Choosing $\gamma^\star$ closer to 1 means that the computed solution on our final mesh comes closer to solving (4.2), but a decrease in $\gamma^\star$ can greatly increase the number of algorithm iterations while producing only a slight improvement in the accuracy of the computed solution. This is demonstrated in Figures 5.2 and 5.3 below.

5.2 Example 1: a linear problem

The first test problem follows [17, 21] by taking (1.1) with

$$b(x) = \frac{4\big(1 + \varepsilon(1+x)\big)}{(1+x)^4}, \tag{5.2a}$$

and

$$f(x) = \frac{-4}{(1+x)^4}\Big[\big(1+\varepsilon(1+x)+4\pi^2\varepsilon^2\big)\cos(2\pi t) - 2\pi\varepsilon^2(1+x)\sin(2\pi t) + 3\big(1+\varepsilon(1+x)\big)\frac{e^{-1/\varepsilon}}{1-e^{-1/\varepsilon}}\Big], \quad (5.2b)$$

where $t = 2x/(x+1)$. The boundary conditions are

$$u(0) = 2, u(1) = -1. \quad (5.2c)$$

The solution to this problem is

$$u(x) = -\cos(2\pi t) + \frac{3(e^{-t/\varepsilon} - e^{-1/\varepsilon})}{1-e^{1/\varepsilon}},$$

and it exhibits a boundary layer near $x = 0$.

Recalling that u^N is the solution of the discrete problem (2.1), we compute the errors

$$E_\varepsilon^N := \max_i |u(x_i) - u_i^N|,$$

where the x_i are the points in the final mesh generated by the algorithm of §5.1. For each N, we shall compute $E^N := \max_\varepsilon E_\varepsilon^N$ over a wide range of values of ε. This is an indicator of the robustness of the algorithm as ε varies. Rates of convergence are computed in the usual way:

$$\rho_\varepsilon^N := \ln\big(E_\varepsilon^N / E_\varepsilon^{2N}\big)/\ln 2.$$

First, consider an implementation of the arc-length monitor function similar to that used for convection-diffusion problems in [8, 9, 20]. Instead of (4.7), take $M_i^{(k)} = 1 + |D^- u_i^N|$. Since $D^- u_i^N$ is an approximation for $u'(x)$ at the mid-point of x_{i-1} and x_i, replace (5.1) by

$$\tilde{M}_i^{(k)} = 1 + |D^- u_i^N| \quad \text{for each } i. \quad (5.3)$$

Table 5.1 displays the errors and the rates of convergence for the computed solutions. The errors are erratic and large relative to those given later for (4.7). In Figure 5.1 we compute E_ε^N for $\varepsilon^2 = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-32}$ and give a log-log plot of E_ε^N as a function of ε for each of $N = 2^7, 2^8, \ldots, 2^{14}$. For any fixed N there is a noticeable increase in the error as ε decreases, except in the evidently unsatisfactory situation where all mesh points fall outside the boundary layer (this happens when ε is very small relative to N^{-1}).

Next, in Table 5.2, the errors in the computed solutions for the monitor function (4.7) are displayed. They are much smaller than those of Table 5.1 and for any fixed N do not vary significantly with ε. Some variation in the results is to be expected since we do not require that the equidistribution problem (4.2) be solved exactly. The rates of convergence ρ_ε^N are given and show that the method is second order.

Table 5.2 also lists the number of iterations k required by the algorithm. We observe that k is clearly proportional to $[\ln(1/\varepsilon)]/\ln N$ when ε is small relative to N; when ε is not small relative to N, a simple equidistant mesh is adequate for computing an accurate solution so adaptivity is unnecessary and the dependence of k on ε and N is consequently unimportant.

In Figure 5.2 we show the errors E_ε^N when $\varepsilon^2 = 1, 2^{-1}, 2^{-2}, \ldots, 2^{-32}$. Each continuous line represents the errors computed for fixed N and $\gamma^\star = 2$. The dashed lines are computed with

ε^2	$N=2^6$	$N=2^7$	$N=2^8$	$N=2^9$	$N=2^{10}$	$N=2^{11}$	$N=2^{12}$
1	3.43e-3	2.93e-3	7.83e-4	2.13e-4	5.43e-5	1.36e-5	3.42e-6
ρ_ε^N	0.23	1.90	1.88	1.97	1.99	2.00	
1e-01	9.25e-3	2.91e-3	8.16e-4	2.12e-4	5.34e-5	1.34e-5	3.35e-6
ρ_ε^N	1.67	1.83	1.95	1.99	2.00	2.00	
1e-02	1.23e-2	5.72e-3	1.57e-3	4.76e-4	1.28e-4	3.27e-5	8.21e-6
ρ_ε^N	1.10	1.86	1.72	1.89	1.97	1.99	
1e-04	4.17e-2	4.69e-3	3.76e-3	2.74e-3	1.06e-3	3.22e-4	8.46e-5
ρ_ε^N	3.15	0.32	0.46	1.37	1.72	1.93	
1e-08	5.23e-3	1.29e-2	1.52e-2	4.35e-3	4.30e-3	2.24e-3	9.34e-4
ρ_ε^N	-1.30	-0.24	1.81	0.02	0.94	1.26	
1e-16	5.53e-2	5.69e-4	1.33e-2	4.36e-3	6.71e-3	2.63e-4	6.90e-4
ρ_ε^N	6.60	-4.55	1.61	-0.62	4.67	-1.39	
E^N	5.53e-2	1.29e-2	1.52e-2	4.36e-3	6.71e-3	2.24e-3	9.34e-4

Table 5.1: Errors and rates of convergence using arc-length monitor function (5.3)

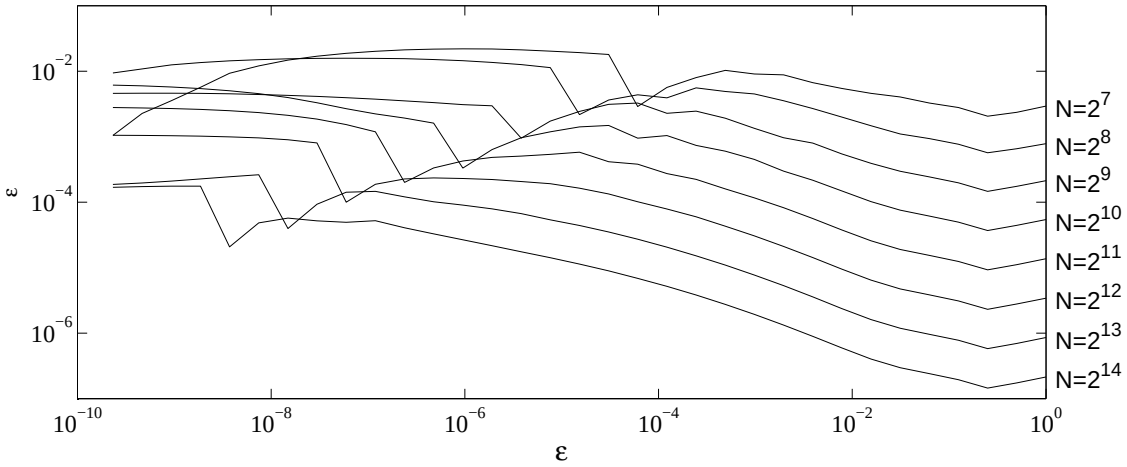

Figure 5.1: Errors in u^N when using arc-length monitor function (5.3) to solve (5.2)

$\gamma^\star = 1.1$. In contrast to Figure 5.1 the error is roughly constant as ε decreases. Although the errors are more uniform for the smaller value of $\gamma^\star$, the improvement is insignificant.

The dependence of the mesh on the choice of $\gamma^\star$ is further demonstrated in Figure 5.3, where we show how a mesh with $N = 32$ intervals evolves through successive iterations of the algorithm. The right-hand part of this figure is a blow-up of the left-hand part near $x = 0$. The value of $\gamma^{(k)}$ is shown for each iteration; from this one can deduce the precise iteration at which a particular choice of $\gamma^\star$ would have caused the algorithm to terminate.

The mesh-generating function for the final mesh computed by the algorithm is given in Figure 5.4 for the case $\varepsilon^2 = 10^{-8}$ and $N = 64$, as are the mesh-generating functions for the corresponding Shishkin [16, Chapter 6] and Bakhvalov meshes (see, e.g., [10, §3.4.1]). It is evident that inside the boundary layer, the final computed mesh strongly resembles a Bakhvalov (but not a Shishkin) mesh. The distribution of the error $|u(x_i) - x_i^N|$ as a function of the mesh index i is shown in Figure 5.5 for these three meshes.

	$N = 2^6$				$N = 2^7$				$N = 2^8$			
ε^2	E_ε^N	ρ_ε^N	k	$k\frac{\ln N}{\ln 1/\varepsilon}$	E_ε^N	ρ_ε^N	k	$k\frac{\ln N}{\ln 1/\varepsilon}$	E_ε^N	ρ_ε^N	k	$k\frac{\ln N}{\ln 1/\varepsilon}$
1	1.81e-3	1.99	1		4.54e-4	1.60	1		1.50e-4	1.95	1	
1e-01	2.60e-3	1.78	1	3.6	7.56e-4	1.96	1	4.2	1.95e-4	1.95	1	4.8
1e-02	2.56e-3	2.12	1	1.8	5.87e-4	1.82	1	2.1	1.66e-4	1.99	1	2.4
1e-04	1.05e-3	2.01	2	1.8	2.61e-4	2.00	1	1.1	6.52e-5	2.00	1	1.2
1e-08	1.56e-3	1.93	4	1.8	4.09e-4	1.99	2	1.1	1.03e-4	2.10	2	1.2
1e-16	1.49e-3	2.04	9	2.0	3.64e-4	2.00	6	1.6	9.09e-5	2.03	5	1.5
E^N	2.60e-3				7.56e-4				1.95e-4			

	$N = 2^9$				$N = 2^{10}$				$N = 2^{11}$			
ε^2	E_ε^N	ρ_ε^N	k	$k\frac{\ln N}{\ln 1/\varepsilon}$	E_ε^N	ρ_ε^N	k	$k\frac{\ln N}{\ln 1/\varepsilon}$	E_ε^N	ρ_ε^N	k	$k\frac{\ln N}{\ln 1/\varepsilon}$
1	3.87e-5	1.98	1		9.84e-6	1.97	1		2.52e-6	1.92	1	
1e-01	5.04e-5	1.95	1	5.4	1.30e-5	1.98	1	6.0	3.29e-6	1.98	1	6.6
1e-02	4.16e-5	2.01	1	2.7	1.04e-5	2.00	1	3.0	2.59e-6	2.00	1	3.3
1e-04	1.63e-5	2.00	1	1.4	4.07e-6	2.00	1	1.5	1.02e-6	2.00	1	1.7
1e-08	2.40e-5	2.17	2	1.4	5.33e-6	2.04	2	1.5	1.29e-6	2.00	2	1.7
1e-16	2.23e-5	2.01	5	1.7	5.53e-6	1.95	4	1.5	1.43e-6	2.01	4	1.7
E^N	5.04e-5				1.30e-5				3.29e-6			

Table 5.2: Errors in u^N applying monitor function (4.7) to problem (5.2) with $\gamma^\star = 2$

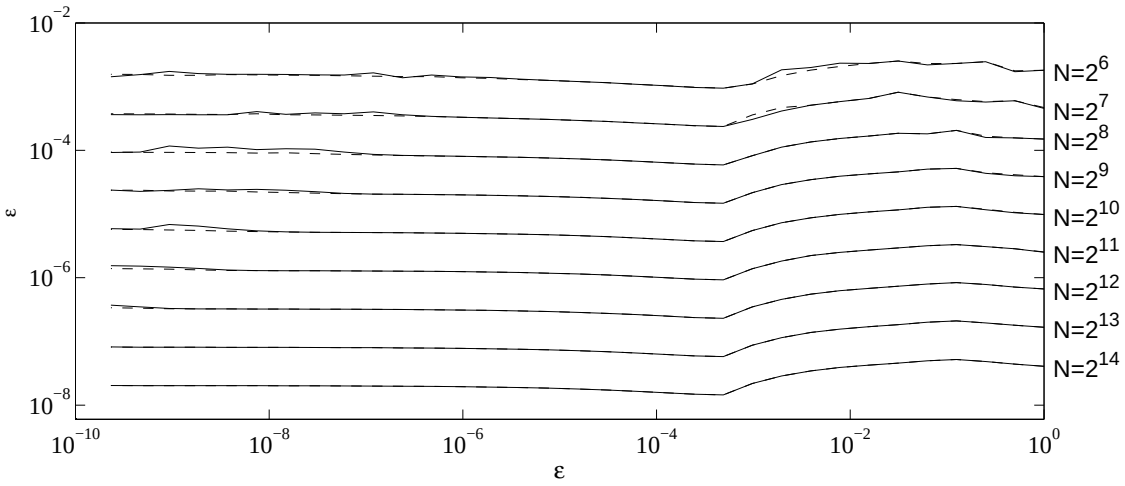

Figure 5.2: As in Table 5.2, but for $\gamma^\star = 2$ (continuous line) and $\gamma^\star = 1.1$ (dashed line)

5.3 A nonlinear problem

To demonstrate that our monitor function and algorithm can be successfully applied in a nonlinear setting, consider the reaction-diffusion problem

$$-\varepsilon u'' + p(x,u) = 0, \qquad u(0) = u(1) = 0. \tag{5.4}$$

Sun and Stynes [23] used a Shishkin mesh to solve this problem, with certain hypotheses on $p(\cdot,\cdot)$ that permitted (5.4) to have multiple solutions. They showed that their numerical method was almost second-order convergent. More recently, the numerical solution of (5.4) on Shishkin and Bakhvalov meshes was studied in [10].

The numerical test problem in [10, 23] is

$$\varepsilon^2 u'' - (u^2 + u - 0.75)(u^2 + u - 3.75) = 0, \qquad u(0) = 0,\ u(0) = 0. \tag{5.5}$$

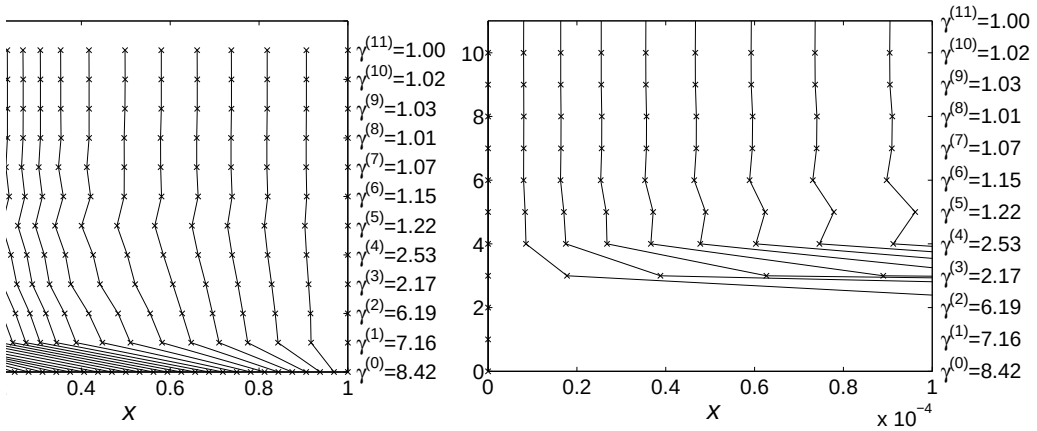

Figure 5.3: Evolution of the mesh when applying (4.7) to (5.2) with $\varepsilon^2 = 10^{-8}, N = 32$

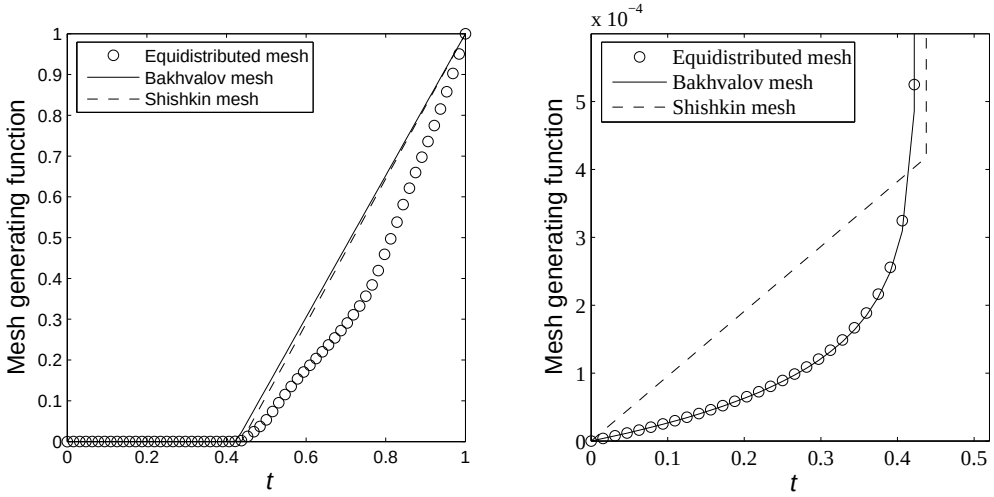

Figure 5.4: Mesh-generating functions for adapted and a priori meshes; $\varepsilon^2 = 10^{-8}$ and $N = 64$.

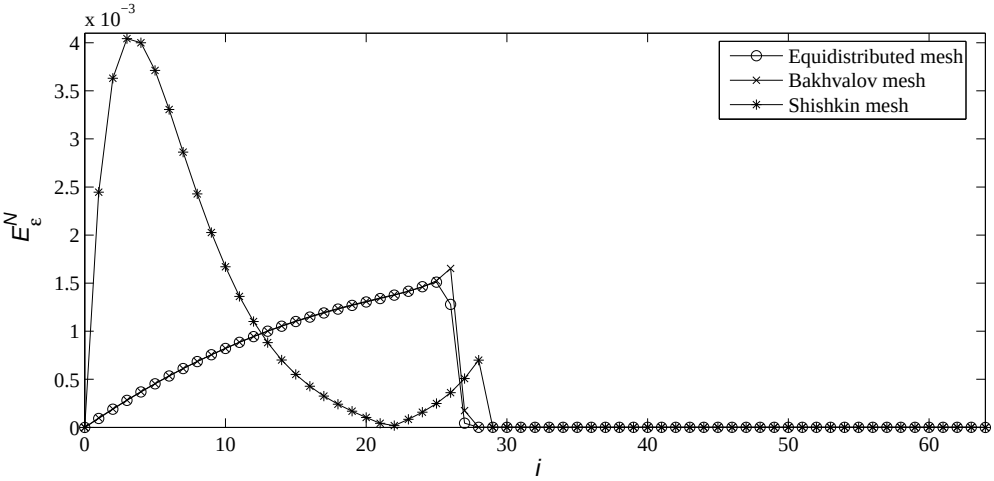

Figure 5.5: Error in u^N as a function of the mesh index i, with $\varepsilon^2 = 10^{-8}$ and $N = 64$.

It has two solutions that are stable in the sense of [23]. We now apply the equidistribution approach to compute an approximation to the stable solution u that, away from the boundary, satisfies $u \approx -1.5$.

The true solution is unknown so we take as our reference solution $U(x)$, the piecewise linear interpolant to the numerical solution to (5.5) computed on a Bakhvalov mesh with $N = 2^{16}$ intervals. See [10, §3.4.1] for details of how to construct this mesh. The error in each computed solution is then

$$E_\varepsilon^N := \max_i |U(x_i) - u_i^N|.$$

Table 5.3 gives the results obtained using the monitor function (4.7), where the expression $|\delta^2(bu)_i)|^{1/2}$ is replaced by $|\delta^2 p(x_i, u_i)|^{1/2}$. The method is quite successful: for larger values of N, the errors are robust with respect to ε, and the convergence rates are close to 2.

	$N = 2^6$		$N = 2^7$		$N = 2^8$		$N = 2^9$		$N = 2^{10}$		$N = 2^{11}$	
ε^2	E_ε^N	k	E_ε^N	k	E_ε^N	k	E_ε^N	k	E_ε^N	k	E_ε^N	k
1	1.09e-5	0	2.72e-6	0	6.81e-7	0	1.70e-7	0	4.26e-8	0	1.07e-8	0
ρ_ε^N	2.00		2.00		2.00		2.00		1.99		1.96	
1e-01	4.20e-4	1	1.03e-4	1	2.54e-5	1	6.33e-6	1	1.58e-6	1	3.97e-7	1
ρ_ε^N	2.03		2.02		2.01		2.00		2.00		1.99	
1e-02	7.11e-4	1	1.93e-4	1	4.79e-5	1	1.19e-5	1	2.98e-6	1	7.49e-7	1
ρ_ε^N	1.88		2.01		2.01		1.99		1.99		1.99	
1e-04	9.25e-4	2	2.23e-4	2	5.53e-5	1	1.40e-5	1	3.50e-6	1	8.77e-7	1
ρ_ε^N	2.05		2.01		1.98		2.00		1.99		1.99	
1e-08	8.57e-4	4	2.26e-4	4	5.69e-5	3	1.42e-5	3	3.60e-6	3	8.94e-7	3
ρ_ε^N	1.92		1.99		2.01		1.98		2.01		1.99	
1e-16	9.44e-4	13	2.18e-4	6	5.65e-5	6	1.42e-5	5	3.59e-6	4	8.94e-7	4
ρ_ε^N	2.11		1.95		2.00		1.98		2.01		1.99	

Table 5.3: Errors and rates of convergence for approximations to (5.5) using (4.7)

6 Conclusions

Numerical experiments show that each of the monitor functions (4.7), (4.8) and (4.9) can be successfully applied to a variety of singularly perturbed reaction-diffusion problems. Among these monitor functions, as (4.7) is the easiest to implement, we recommend it as the best choice for reaction-diffusion problems in general.

Acknowledgement. We are grateful to John Mackenzie for several useful comments on an earlier version of this paper.

References

[1] V.B. Andreev, On the uniform convergence on arbitrary meshes of the classical difference scheme for a one-dimensional singularly perturbed reaction-diffusion problem, Ž. Vyčisl. Mat. i Mat. Fiz. 3 (2004) 474–489 (in Russian).

[2] N. S. Bakhvalov, Towards optimization of methods for solving boundary value problems in the presence of a boundary layer, Zh. Vychisl. Mat. Mat. Fiz., 9 (1969), 841–859 (in Russian).

[3] G. Beckett and J. A. Mackenzie, Convergence analysis of finite difference approximations to a singularly perturbed boundary value problem, Appl. Numer. Math., 35 (2000) 87–109.

[4] G. Beckett and J. A. Mackenzie, On a uniformly accurate finite difference approximation of a singularly perturbed reaction-diffusion problem using grid equidistribution, J. Comput. Appl. Math., 131 (2001) 381–405.

[5] W. Cao, W. Huang and R. D. Russell, A study of monitor functions for two-dimensional adaptive mesh generation, SIAM J. Sci. Comput. 20 (1999) 1978–1994.

[6] T.-F. Chen and H.-D. Yang, Numerical construction of optimal adaptive grids in two spatial dimensions, Computers Math. Applic. 39 (2000) 101–120.

[7] W. Huang and D. M. Sloan, A simple adaptive grid in two dimensions, SIAM J. Sci. Comput. 15 (1994) 776–797.

[8] N. Kopteva, Maximum norm a posteriori error estimates for a one-dimensional convection-diffusion problem, SIAM J. Numer. Anal., 39 (2001) 423–441.

[9] N. Kopteva and M. Stynes, A robust adaptive method for a quasilinear one-dimensional convection-diffusion problem, SIAM J. Numer. Anal. 39 (2001) 1446–1467.

[10] N. Kopteva and M. Stynes, Numerical analysis of a singularly perturbed nonlinear reaction-diffusion problem with multiple solutions, Appl. Numer. Math. 51 (2004) 273–288.

[11] H.-O. Kreiss, T.A. Manteuffel, B. Swartz, B. Wendroff and A.B. White, Supra-convergent schemes on irregular grids, Math. Comp. 47 (1986) 537–554.

[12] T. Linß, Uniform pointwise convergence of finite difference schemes using grid equidistribution, Computing 66 (2001) 27–39.

[13] T. Linß and N. Madden, A finite element analysis of a coupled system of singularly perturbed reaction-diffusion equations, Appl. Math. Comput. 148 (2004) 869–880.

[14] J. A. Mackenzie, Uniform convergence analysis of an upwind finite-difference approximation of a convection-diffusion boundary value problem on an adaptive grid, IMA J. Numer. Anal. 19 (1999) 233–249.

[15] N. Madden and M. Stynes, A uniformly convergent numerical method for a coupled system of two singularly perturbed linear reaction-diffusion problems, IMA J. Numer. Anal. 23 (2003) 627–644.

[16] J. J. H. Miller, E. O'Riordan and G. I. Shishkin, Solution of Singularly Perturbed Problems with ε-uniform Numerical Methods — Introduction to the Theory of Linear Problems in One and Two Dimensions, World Scientific, Singapore, 1996.

[17] E. O'Riordan and M. Stynes, A uniformly accurate finite element method for a singularly perturbed one-dimensional reaction-diffusion problem, Math. Comp. 47 (1986) 555–570.

[18] Y. Qiu and D. M. Sloan, Analysis of difference approximations to a singularly perturbed two-point boundary value problem on an adaptive grid, J. Comput. Appl. Math. 101 (1999) 1–25.

[19] H.-G. Roos, M. Stynes and L. Tobiska, Numerical Methods for Singularly Perturbed Differential Equations, Volume 24, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1996.

[20] Y. Qiu, D. M. Sloan and T. Tang, Numerical solution of a singularly perturbed two-point boundary value problem using equidistribution: analysis of convergence, J. Comput. Appl. Math. 116 (2000) 121–143.

[21] A.H. Schatz and L.B. Wahlbin, On the finite element method for singularly perturbed reaction-diffusion problems in two and one dimensions, Math. Comp. 40 (1983) 47–88.

[22] M. Stynes and L. Tobiska, A finite difference analysis of a streamline diffusion method on a Shishkin mesh, Numer. Algorithms 18 (1998) 337–360.

[23] G. Sun and M. Stynes, A uniformly convergent method for a singularly perturbed semilinear reaction-diffusion problem with multiple solutions, Math. Comput. 65 (1996) 1085–1109.

[24] A.N. Tihonov and A.A. Samarskiĭ, Homogeneous difference schemes on irregular meshes, Ž. Vyčisl. Mat. i Mat. Fiz. (1962) 812–832 (in Russian).