fractal and fractional

MDPI

Article

A Second-Order Adaptive Grid Method for a Singularly Perturbed Volterra Integrodifferential Equation

Libin Liu [1,2], Ying Liang [2] and Yong Zhang [3,*]

[1] Guangxi Key Laboratory of Human-Machine Interaction and Intelligent Decision, Nanning Normal University, Nanning 530100, China
[2] School of Mathematics and Statistics, Nanning Normal University, Nanning 530010, China
[3] School of Big Data and Artificial Intelligence, Chizhou University, Chizhou 247000, China
* Correspondence: yongzhang@czu.edu.cn

Abstract: In this paper, an adaptive grid method for a singularly perturbed Volterra integro-differential equation is studied. Firstly, this problem is discretized by a new second-order finite difference scheme, for which a truncation error analysis is conducted. Then, based on this truncation error bound and the mesh equidistribution principle, we show that there is a mesh that provides an optimal error bound of $O(N^{-2})$, which is robust with respect to the perturbation parameter. Finally, based on an approximation monitor function, an adaptive grid generation algorithm is constructed and some numerical results are given to support our theoretical results.

Keywords: singularly perturbed; Volterra integro-differential equation; uniformly convergent; adaptive grid method

Citation: Liu, L.; Liang, Y.; Zhang, Y. A Second-Order Adaptive Grid Method for a Singularly Perturbed Volterra Integrodifferential Equation. *Fractal Fract.* **2022**, *6*, 636. https://doi.org/10.3390/fractalfract6110636

Academic Editors: Xinguang Zhang, Yonghong Wu, Chuanjun Chen, Jiwei Zhang, Chun Lu and Haci Mehmet Baskonus

Received: 08 October 2022
Accepted: 26 October 2022
Published: 1 November 2022

1. Introduction

In this paper, we consider an adaptive grid method for the following singularly perturbed Volterra integrodifferential equation:

$$\begin{cases} Lu(x) := \varepsilon u'(x) + a(x)u(x) + \int_0^x K(x,s)u(s)ds = f(x), \ x \in (0,1], \\ u(0) = A, \end{cases} \tag{1}$$

where $0 < \varepsilon \ll 1$, A is a given constant, and $a(x)$, $K(x,s)$, and $f(x)$ are sufficiently smooth functions, which are independent of the parameter ε. It is assumed that there exists a positive constant α such that $a(x) \geq \alpha > 0$. Under these conditions, the problem (1) has a unique solution $u(x)$ (see [1,2]), which typically exhibits a boundary layer at $x = 0$ as $\varepsilon \to 0$. Thus, this type of problem is called singularly perturbed Volterra integrodifferential equations (SPVIDEs), which often appear in physics [3], biology [4,5], and other areas [6].

Due to the presence of this perturbation parameter, standard finite difference and finite-element methods on a uniform mesh for (1) may yield inaccurate numerical results. Therefore, some special numerical methods for SPVIDEs have been discussed in studies such as [1,7–14]. Among these methods, the authors in [7,10] developed some layer-adapted mesh methods to solve SPVIDEs. Especially, Yanman and Amiraliyev [1] proposed a new discretization scheme for problem (1), which was almost second-order uniform convergent on the Shishkin mesh.

In addition to the layer-adaptive grid method mentioned above, the adaptive grid method for singularly perturbed convection-diffusion problems has also attracted much attention over the last decade, see [15,16] and the monograph [17]. For the adaptive grid methods of SPVIDEs, Sumit et.al. [18] proposed a first-order uniform convergent adaptive grid method for a nonlinear singularly perturbed Volterra integro-differential equation . To the best of our knowledge, there are few second-order adaptive grid methods for SPVIDEs

except for the Richardson extrapolation technique given in [19]. Based on the discretization scheme proposed in [1], the authors in [20] developed an adaptive grid algorithm for a singularly perturbed convection-diffusion equation. However, they did not carry out a convergence analysis. Thus, it is very desirable to construct a second-order adaptive grid method (not a hybrid discretization scheme) for singularly perturbed problems.

Motivated by [1,20], the aim of this paper is to design a new second-order adaptive grid method for a singularly perturbed Volterra integro-differential equation. By virtue of a truncation error analysis, a suitable monitor function is chosen to design an adaptive grid based on the mesh equidistribution principle. Furthermore, we prove that our presented adaptive grid method is second-order uniform-convergent with respect to the perturbation parameter. Finally, several numerical results are given to confirm our theoretical findings.

Notation. Throughout the paper, C denotes a generic positive constant that is independent of ε and the mesh parameter N. It may take different values in different places. For any continuous function $g(x)$, we use the notation $g_i = g(x_i)$ and $\|g\|_\infty = \max\limits_{x\in[0,1]} |g(x)|$.

2. Preliminary Results

We provide the following bounds for the derivatives of $u(x)$ (see [21,22]), which will be used in later analysis.

Lemma 1. *Assuming $K(x,s)$, $f(x)$, $a(x)$ are sufficiently smooth functions and there is a positive constant α such that $a(x) \geq \alpha > 0$, then the solution $u(x)$ of problem (1) has the following bounds:*

$$|u^{(k)}(x)| \leq C\left(1 + \frac{1}{\varepsilon^k} e^{-\frac{\alpha x}{\varepsilon}}\right), x \in [0,1], k = 0,1,2. \tag{2}$$

To obtain our numerical discretization scheme, we consider an arbitrary nonuniform mesh $\overline{\Omega}^N \equiv \{0 = x_0 < x_1 < \cdots < x_N = 1\}$, where N is a positive integer. For $i = 1, \cdots, N$, $h_i = x_i - x_{i-1}$ represents the local mesh step.

Similar to reference [1], we provide the construction of our numerical scheme on $\overline{\Omega}$. By multiplying both sides of the first equation of (1) by $\varphi_i(x) = e^{-\frac{a_i(x_i - x)}{\varepsilon}}$ and integrating on the interval $[x_{i-1}, x_i]$, then multiplying both sides by $\chi_i^{-1} h_i^{-1}$, the origin problem (1) can be written into the following integral equation:

$$\chi_i^{-1} h_i^{-1} \int_{x_{i-1}}^{x_i} Lu\varphi_i(x)dx = \chi_i^{-1} h_i^{-1} \int_{x_{i-1}}^{x_i} f(x)\varphi_i(x)dx, \tag{3}$$

where $\chi_i = h_i^{-1} \int_{x_{i-1}}^{x_i} \varphi_i(x)dx = \frac{1 - e^{-a_i\rho_i}}{a_i\rho_i}$ and $\rho_i = \frac{h_i}{\varepsilon}$ for $i = 1 \cdots, N$.

Furthermore, for the differential part of Lu defined in (1), it follows from the left side of Equation (3) that

$$\begin{aligned} \chi_i^{-1} h_i^{-1} \int_{x_{i-1}}^{x_i} \left[\varepsilon u'(x) + a(x)u(x)\right]\varphi_i(x)dx &= \varepsilon\theta_i D^- u_i + a_i u_i \\ &+ \chi_i^{-1} h_i^{-1} \int_{x_{i-1}}^{x_i} [a(x) - a(x_i)]\varphi_i(x)u(x)ds, \end{aligned} \tag{4}$$

where $\theta_i = \frac{a_i\rho_i e^{-a_i\rho_i}}{1 - e^{-a_i\rho_i}}$ and $D^- u_i = \frac{u_i - u_{i-1}}{h_i}$.

Next, by using the Newton interpolating formula to $a(x)$ at points x_{i-1},x_i and substituting it into Equation (4), we have

$$\chi_i^{-1} h_i^{-1} \int_{x_{i-1}}^{x_i} \left[\varepsilon u'(x) + a(x)u(x)\right]\varphi_i(x)dx = \varepsilon\theta_i D^- u_i + \left(a_i + h_i\delta_i D^- a_i\right)u_i + R_i^{(1)}, \tag{5}$$

where $\delta_i = \frac{a_i\rho_i}{1-e^{-a_i\rho_i}} - \frac{1}{a_i\rho_i}$, $a_i = a(x_i)$ and

$$\begin{aligned} R_i^{(1)} &= \chi_i^{-1}h_i^{-1}\int_{x_{i-1}}^{x_i}\frac{a''(\eta_i)}{2}(x-x_{i-1})(x-x_i)\varphi_i(x)u(x)dx \\ &\quad - a_{\bar{x},i}\chi_i^{-1}h_i^{-1}\int_{x_{i-1}}^{x_i}(x-x_i)\varphi_i(x)\left(\int_x^{x_i}u'(s)ds\right)dx,\ \eta_i\in[x_{i-1},x_i]. \end{aligned} \tag{6}$$

Similarly, the right hand of (3) can be written as follows:

$$\chi_i^{-1}h_i^{-1}\int_{x_{i-1}}^{x_i}f(x)\varphi_i(x)dx = \bar{f}_i + R_i^{(2)}, \tag{7}$$

where $\bar{f}_i = f_i + h_i\delta_i D^- f_i$ and

$$R_i^{(2)} = \frac{1}{2}\chi_i^{-1}h_i^{-1}\int_{x_{i-1}}^{x_i}f''(\gamma_i)(x-x_{i-1})(x-x_i)\varphi_i(x)dx,\ \gamma_i\in[x_{i-1},x_i]. \tag{8}$$

Meanwhile, for the integral part of Lu, by using the trapezoidal formula with basis function $\varphi_i(x)$ and remainder term in integral form, we obtain

$$\begin{aligned} &\chi_i^{-1}h_i^{-1}\int_{x_{i-1}}^{x_i}\varphi_i(x)\left(\int_0^x K(x,s)u(s)ds\right)dx \\ &\quad = \sum_{j=0}^{i}\hbar_j\kappa_{i,j}u_j + R_i^{(3)} + R_i^{(4)}, \end{aligned} \tag{9}$$

where

$$\begin{aligned} &\kappa_{i,j} = \kappa(x_i,x_j) = K(x_i,x_j) + h_i\delta_i\frac{\partial K}{\partial x}(x_i,x_j), \\ &\hbar_0 = \frac{h_1}{2}, \hbar_j = \frac{h_j+h_{j+1}}{2}, j = 1,\cdots,N-1, \hbar_N = \frac{h_N}{2}, \\ &R_i^{(3)} = \chi_i^{-1}h_i^{-1}\int_{x_{i-1}}^{x_i}dx\varphi_i(x)\int_{x_{i-1}}^{x_i}\frac{d^2}{d\xi^2}\left(\int_0^\xi K(\xi,s)u(s)ds\right)(\xi-x)d\xi, \\ &R_i^{(4)} = \frac{1}{2}\sum_{j=1}^{i}\int_{x_{j-1}}^{x_j}(x_j-\xi)(x_{j-1}-\xi)\frac{d^2}{d\xi^2}\kappa(x_i,\xi)u(\xi)d\xi. \end{aligned}$$

Finally, neglecting the truncation errors given in (5), (7), and (9), we obtain the finite difference scheme of problem (1) as follows:

$$\begin{cases} L^N u_i^N \equiv \varepsilon\theta_i D^- u_i^N + \bar{a}_i u_i^N + \sum\limits_{j=0}^{i}\hbar_j\kappa_{i,j}u_j^N = \bar{f}_i,\ 1\le i\le N, \\ u_0^N = A, \end{cases} \tag{10}$$

where u_i^N is an approximation solution of $u(x)$ at $x = x_i$ and $\bar{a}_i = a_i + (D^- a_i + K_{ii})h_i\delta_i$.

Next, we provide a lemma (see lemma 4.1 in [22]), which will be used in the proof of Lemma 3.

Lemma 2. *Consider the following difference problem:*

$$\ell_N v_i = \varepsilon v_{\bar{t},i} + a_i v_i = F_i, i = 0,1,2,\cdots,N_0. \tag{11}$$

$$v_0 = A. \tag{12}$$

Let $|F_i| \leq \mathcal{F}_i$ and $\mathcal{F}_i$ be nondecreasing function. Then the solution of (11) and (12) satisfies

$$|v_i| \leq |A| + \alpha^{-1}\mathcal{F}_i, i = 0, 1, 2, \cdots, N_0. \tag{13}$$

In order to derive the convergence result of the numerical solution $\{u_i^N\}_{i=0}^N$, we provide the following stability result.

Lemma 3. *Assume that there is a constant α_* such that $\bar{a}_i + \hbar_i\kappa(x_i, x_i) \geq \alpha_* > 0$, $i = 1, \ldots, N$. Then, we have*

$$\max_{0\leq i\leq N}\left|u_i^N\right| \leq C\left(\max_{0\leq i\leq N}|\bar{f}_i| + A\right). \tag{14}$$

Proof. For each u_i^N, $i = 1, \cdots, N$, we first define the following difference operator:

$$\ell^N u_i^N := \varepsilon\theta_i D^- u_i^N + (\bar{a}_i + \hbar_i\kappa_{i,i})u_i^N. \tag{15}$$

Then, by using Lemma 2, we have

$$|u_i^N| \leq \alpha_*^{-1}\left(|u_0^N| + |\ell^N u_i^N|\right). \tag{16}$$

It follows from the first equation of (10) that

$$\ell^N u_i^N = L^N u_i^N - \sum_{j=0}^{i-1}\hbar_j\kappa_{i,j}u_j^N. \tag{17}$$

Furthermore, since $\kappa(x, s)$ is bounded, it yields

$$|\ell^N u_i^N| \leq |\bar{f}_i| + C\sum_{j=0}^{i-1}\hbar_j|u_j^N|, \ 1 \leq i \leq N. \tag{18}$$

Combining with (16), we have

$$|u_i^N| \leq \left(|u_0^N| + \alpha_*^{-1}\|f\|_\infty\right) + \alpha_*^{-1}\left(C\sum_{j=1}^{i-1}\hbar_j|u_j^N|\right). \tag{19}$$

Finally, applying Gronwall's inequality to (19) yields

$$\max_{0\leq i\leq N}\left|u_i^N\right| \leq (A + \alpha_*^{-1}\|f\|_\infty)\exp\left(\alpha_*^{-1}C\sum_{j=1}^{i-1}\hbar_j\right), \tag{20}$$

which completes the proof. □

3. Truncation Error Analysis

Let $z_i = u_i^N - u_i$ be the error at x_i in the computed solution. Then,

$$\begin{cases} L^N z_i = R_i, \ i = 1, \cdots, N, \\ z_0 = 0, \end{cases} \tag{21}$$

where $R_i = R_i^{(1)} + R_i^{(2)} + R_i^{(3)} + R_i^{(4)}$ is the local truncation error at nodal x_i.

Lemma 4. *Assuming $K(x,s)$, $f(x)$, $a(x)$ are sufficiently smooth functions and there is a positive constant α such that $a(x) \geq \alpha > 0$, the truncation error R_i has the following bound:*

$$\max_{1\leq i\leq N}|R_i| \leq C\left[\max_{1\leq i\leq N} h_i^2 + \max_{1\leq i\leq N} h_i \int_{x_{i-1}}^{x_i} |u'(x)|dx + \max_{1\leq j\leq N}\left(\int_{x_{j-1}}^{x_j} \varepsilon^{-1}e^{-\frac{\alpha x}{2\varepsilon}}dx\right)^2\right].$$

Proof. From Lemma 3.1 of [1], we can easily obtain

$$|R_i^{(1)}| + |R_i^{(2)}| + |R_i^{(3)}| \leq Ch_i\left(h_i + \int_{x_{i-1}}^{x_i} |u'(x)|dx\right). \tag{22}$$

For $R_i^{(4)}$, we have

$$\begin{aligned}
|R_i^{(4)}| &\leq C\left(\sum_{j=1}^{i} h_j^3 + \sum_{j=1}^{i}\int_{x_{j-1}}^{x_j}(x_j-\xi)(\xi-x_{j-1})\left|u'(\xi)+u''(\xi)\right|d\xi\right)\\
&\leq C\max_{1\leq j\leq N} h_j^2 + \max_{1\leq j\leq N}\int_{x_{j-1}}^{x_j}|u'(\xi)|(\xi-x_{j-1})d\xi\\
&\quad + \max_{1\leq j\leq N}\int_{x_{j-1}}^{x_j}\varepsilon^{-2}e^{-\frac{\alpha x}{\varepsilon}}(\xi-x_{j-1})d\xi\\
&\leq C\max_{1\leq j\leq N} h_j^2 + \max_{1\leq j\leq N} h_j\int_{x_{j-1}}^{x_j}|u'(x)|dx + \max_{1\leq j\leq N}\left(\int_{x_{j-1}}^{x_j}\varepsilon^{-1}e^{-\frac{\alpha x}{2\varepsilon}}dx\right)^2,
\end{aligned} \tag{23}$$

where we have used the fact that

$$\int_a^b \phi(s)(s-a)ds \leq \frac{1}{2}\left\{\int_a^b \phi(s)^{1/2}ds\right\}^2 \tag{24}$$

holds true for any positive monotonically decreasing function $\phi(s)$ on $[a,b]$. Furthermore, combining (22) and (23), we complete the proof of this lemma. □

4. Adaptive Grid and Convergence Analysis

Monitor functions are widely used by many researchers (see, e.g., [18,23–27]) to design an adaptive grid algorithm that produces layer-resolving meshes in solving singularly perturbed problems. As is stated in [26], if the monitor functions contain the exact solution of the considered problem, these approaches are called semi-discretization adaptive grid methods. For this purpose, we also study the semi-discretization adaptive grid method for problem (1). Based on the truncation error estimation given in Lemma 4, we choose the following monitor function $M(x,u(x))$:

$$M(x,u(x)) = 1 + |u'(x)| + \varepsilon^{-1}e^{-\frac{\alpha x}{2\varepsilon}}, \tag{25}$$

which is used to construct a grid $\{x_i\}_{i=0}^N$ satisfying

$$\int_{x_{j-1}}^{x_j} M(x,u(x))dx = \frac{1}{N}\int_0^1 M(x,u(x))dx, \quad i = 1,2,\cdots,N. \tag{26}$$

Here, Equation (26) is called the mesh equidistribution principle. It is worth noting that the existence of the grid $\{x_i\}_{i=0}^N$ satisfying (26) can be found in [15], Theorem 3.1.

Lemma 5. *Let $\overline{\Omega}^N = \{x_i\}_{i=0}^N$ be a grid satisfying (26). Then, for $i = 1, \ldots, N$, we have*

$$h_i \le CN^{-1}, \tag{27}$$

$$\int_{x_{i-1}}^{x_i} |u'(x)|dx \le CN^{-1}, \tag{28}$$

$$\int_{x_{i-1}}^{x_i} \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}} dx \le CN^{-1}. \tag{29}$$

Proof. It follows from Lemma 1 that

$$\begin{aligned} &\int_0^1 \left(1 + |u'(x)| + \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}}\right) dx \\ &\le C \int_0^1 \left(1 + \varepsilon^{-1} e^{-\frac{\alpha x}{\varepsilon}} + \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}}\right) dx \\ &\le C \int_0^1 \left(1 + \varepsilon^{-1} e^{-\frac{\alpha x}{\varepsilon}}\right) dx \\ &= C\left(1 + \frac{1}{\alpha}\left(1 - e^{-\frac{\alpha}{\varepsilon}}\right)\right) \\ &\le C. \end{aligned} \tag{30}$$

Then, based on the mesh equidistribution principle (26), we have

$$\begin{aligned} h_i &= x_i - x_{i-1} \\ &\le \int_{x_{i-1}}^{x_i} \left(1 + |u'(x)| + \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}}\right) dx \\ &= \frac{1}{N} \int_0^1 \left(1 + |u'(x)| + \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}}\right) dx \\ &\le CN^{-1}. \end{aligned} \tag{31}$$

Furthermore, by (25) and (26), one has

$$\begin{aligned} \int_{x_{i-1}}^{x_i} |u'(x)|dx &< \int_{x_{i-1}}^{x_i} \left(1 + |u'(x)| + \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}}\right) dx \\ &\le \tfrac{1}{N} \int_0^1 \left(1 + |u'(x)| + \varepsilon^{-1} e^{-\frac{\alpha x}{2\varepsilon}}\right) dx \\ &\le CN^{-1}. \end{aligned} \tag{32}$$

Similarly, we can prove (29). The proof is completed. □

Finally, based on the above preliminary results, we can derive the main theorem about the convergence analysis of presented scheme (10) on an adaptive grid $\bar{\Omega}^N$.

Theorem 1. *Let u_i be the exact solution of problem (1) and u_i^N be the solution of (10) at the adaptive grid $\overline{\Omega}^N = \{x_i\}_{i=0}^N$ satisfying (26). Then, we have*

$$\max_{0 \le i \le N} \left| u_i^N - u(x_i) \right| \le CN^{-2}. \tag{33}$$

Proof. Firstly, applying Lemma 3 to (21) yields

$$\max_{0 \le i \le N} \left| u_i^N - u(x_i) \right| \le C \max_{1 \le i \le N} |R_i|. \tag{34}$$

Then, it follows from Lemmas 4 and 5 that

$$\begin{aligned}\max_{0\le i\le N}\left|u_i^N-u(x_i)\right| &\le C\left[\max_{1\le i\le N}h_i^2+\max_{1\le i\le N}h_i\int_{x_{i-1}}^{x_i}|u'(x)|dx\right.\\ &\quad\left.+\max_{1\le i\le N}\left(\int_{x_{i-1}}^{x_i}\varepsilon^{-1}e^{-\frac{\alpha x}{2\varepsilon}}dx\right)^2\right]\\ &\le CN^{-2},\end{aligned}\tag{35}$$

which completes the proof. □

5. Numerical Results and Discussion

In Section 5.1, we shall first provide a grid generation algorithm based on the equidistribution of the monitor function (25). Numerical results and discussion are presented by two test examples in Section 5.2.

5.1. Mesh Generation Algorithm

Since the monitor function (25) includes the first-order derivative of the exact solution $u(x)$, it is difficult to obtain an adaptive grid $\{x_i\}_{i=0}^N$ by equidistributing the monitor function (25). In practical computation, we choose the following approximating monitor function

$$\widetilde{M}_i = 1+|D^-u_i^N|+\varepsilon^{-1}e^{-\frac{\alpha x_i}{2\varepsilon}},\ i=1,\cdots,N.\tag{36}$$

Therefore, the key problem of our adaptive grid method is to find $\left\{\left(x_i,u_i^N\right)\right\}_{i=0}^N$, with u_i^N calculated from the discretization scheme ref. to Equation (10) on an adaptive grid $\{x_i\}_{i=0}^N$, such that

$$h_i\widetilde{M}_i=\frac{1}{N}\sum_{j=1}^{N}h_j\widetilde{M}_j\qquad\text{for}\quad i=1,\cdots,N.\tag{37}$$

Finally, in order to obtain a grid $\{x_i\}_{i=0}^N$ and the corresponding numerical solution $\left\{u_i^N\right\}_{i=0}^N$ satisfying (37), we provide the following specific mesh-generation Algorithm 1, which is similar to the algorithm given in [25], Section 5.1.

Algorithm 1: Adaptive grid algorithm

Step 1. Provide an initial uniform mesh $\overline{\Omega}^{N,(0)} = \left\{x_i^{(0)}\right\}_{i=0}^{N}$ with N mesh intervals. Choose a constant $C_0 > 1$ that controls the algorithm terminates.

Step 2. For a given grid $\overline{\Omega}^{N,(k)} = \left\{x_i^{(k)}\right\}_{i=0}^{N}$, $k = 0, 1, \cdots$ and the corresponding computed solution $\left\{u_i^{N,(k)}\right\}_{i=0}^{N}$, set $h_i^{(k)} = x_i^{(k)} - x_{i-1}^{(k)}$ for each i and $\Phi_0^{(k)} = 0$ and $\Phi_i^{(k)} = \sum_{j=1}^{i} h_j^{(k)} \widetilde{M}_j^{(k)}$ for $i = 1, \cdots, N$.

Step 3. Define $C^{(k)} := \frac{N}{\Phi_N^{(k)}} \max\limits_{1 \le i \le N} h_i^{(k)} \widetilde{M}_i^{(k)}$. If $C^{(k)} \le C_0$ holds true, then go to *Step 5*. Otherwise go to *Step 4*.

Step 4. For $i = 0, 1, \cdots, N$, let $Y_i^{(k)} = i\Phi_N^{(k)}/N$ and $\phi^{(k)}(s)$ be a linear interpolation function through knots $\left(\Phi_i^{(k)}, x_i^{(k)}\right)$. Then, generate a new mesh $\overline{\Omega}^{N,(k+1)} = \left\{x_i^{(k+1)}\right\}_{i=0}^{N}$ by $x_i^{(k+1)} = \phi^{(k)}\left(Y_i^{(k)}\right)$ for $i = 0, 1, \cdots, N$. Let $k = k + 1$ and return to *Step 2*.

Step 5. Take $\overline{\Omega}^{N,*} = \overline{\Omega}^{N,(k+1)}$ as the final calculation mesh and $\left\{u_i^{N,*}\right\}_{i=0}^{N} = \left\{u_i^{N,(k+1)}\right\}_{i=0}^{N}$ as the corresponding numerical solution. Then, stop iteration process.

5.2. Numerical Experiments and Discussion

Example 1. The first test problem follows [1] is given by:

$$\varepsilon u' + 2u - \int_0^x (x - s)e^{1-xs}u(s)ds = e^x - x, \quad x \in (0, 1],$$
$$u(0) = 1.$$

Since the exact solution of this problem is not available, the maximum errors and the convergence rates can be evaluated as follows:

$$E_\varepsilon^N = \max_{0 \le i \le N} |u_i^N - u_i^{2N}|, \tag{38}$$

$$r_\varepsilon^N = \log_2\left(\frac{E_\varepsilon^N}{E_\varepsilon^{2N}}\right), \tag{39}$$

where u_i^N is the numerical solution calculated on an adaptive grid $\overline{\Omega}^N = \{x_i\}_{i=0}^N$ and u_i^{2N} is the corresponding approximate solution on the mesh $\overline{\Omega}^{2N}$, which is defined by

$$\overline{\Omega}^{2N} = \left\{x_{\frac{i}{2}} : i = 0, 1, \cdots, 2N\right\} \quad \text{with} \quad x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2}, \ x_i \in \overline{\Omega}^N, \ i = 0, 1, \cdots, N-1.$$

Here, we choose $C_0 = 1.5$, $\alpha = 1$ and apply the presented adaptive grid method to solve Example 1 with different values of ε and N. The errors and rates of convergence for the numerical solution are displayed in Table 1. Meanwhile, in order to illustrate the computational efficiency of our presented adaptive grid algorithm, Table 1 also lists the number of iterations *Iter*. Furthermore, to compare the performance of the presented adaptive mesh with the Shishkin mesh (S-Mesh) and the method given in [20], some numerical results are given in Table 2. The numerical results of Shishkin mesh approach is come from [1].

Table 1. Numerical results of our presented adaptive grid method for Example 1.

ε		$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
10^{-1}	E_ε^N	1.69×10^{4}	4.37×10^{-5}	1.11×10^{-5}	2.80×10^{-6}	7.04×10^{-7}	1.76×10^{-7}
	r_ε^N	1.95	1.98	1.99	1.99	2.00	-
	Iter	1	1	1	1	1	1
10^{-2}	E_ε^N	1.74×10^{-4}	5.77×10^{-5}	1.68×10^{-5}	4.32×10^{-6}	1.13×10^{-6}	2.90×10^{-7}
	r_ε^N	1.60	1.78	1.96	1.93	1.97	-
	Iter	2	2	1	1	1	1
10^{-3}	E_ε^N	1.38×10^{-4}	3.99×10^{-5}	1.16×10^{-5}	3.63×10^{-6}	1.61×10^{-6}	3.45×10^{-7}
	r_ε^N	1.79	1.78	1.69	1.64	1.75	-
	Iter	3	2	2	2	2	1
10^{-4}	E_ε^N	1.30×10^{-4}	3.73×10^{-5}	1.00×10^{-5}	2.53×10^{-6}	6.90×10^{-7}	1.91×10^{-7}
	r_ε^N	1.80	1.89	1.99	1.87	1.85	-
	Iter	3	3	2	2	3	2
10^{-5}	E_ε^N	1.29×10^{-4}	3.84×10^{-5}	9.61×10^{-6}	2.51×10^{-6}	6.33×10^{-7}	1.59×10^{-7}
	r_ε^N	1.75	1.99	1.94	1.99	1.99	-
	Iter	4	4	3	3	2	2
10^{-6}	E_ε^N	1.39×10^{-4}	3.56×10^{-5}	9.85×10^{-6}	2.44×10^{-6}	6.28×10^{-7}	1.58×10^{-7}
	r_ε^N	1.97	1.85	2.01	1.96	1.99	-
	Iter	5	4	4	3	3	3

Table 2. Comparison of numerical results with the other methods for Example 1.

N	$\varepsilon = 2^{-12}$			$\varepsilon = 2^{-24}$		
	S-Mesh [1]	**Method [20]**	**Our Method**	**S-Mesh [1]**	**Method [20]**	**Our Method**
64	4.90×10^{-2}	0.4×10^{-4}	1.31×10^{-4}	5.52×10^{-2}	0.4×10^{-4}	1.36×10^{-4}
	1.81	1.92	1.74	1.82	2.03	1.95
128	1.37×10^{-2}	1.05×10^{-5}	3.94×10^{-5}	1.56×10^{-2}	0.97×10^{-5}	3.53×10^{-5}
	1.84	1.87	2.01	1.86	1.97	1.95
256	3.90×10^{-3}	2.85×10^{-6}	9.80×10^{-6}	4.31×10^{-3}	2.47×10^{-6}	9.11×10^{-6}
	1.93	1.79	1.84	1.93	2.04	1.88
512	1.02×10^{-3}	0.82×10^{-6}	2.74×10^{-6}	1.13×10^{-3}	0.6×10^{-6}	2.48×10^{-6}
	1.98	1.65	1.81	1.99	1.99	2.07
1024	2.60×10^{-4}	2.62×10^{-7}	7.81×10^{-7}	2.84×10^{-4}	1.51×10^{-7}	5.91×10^{-7}

It can be observed from Table 1 that the numerical results obtained by the presented adaptive grid method has high accuracy and second-order convergence rate, which supports the theoretical result given in Theorem 1. Moreover, it is shown from the number of iteration *Iter* that the above grid generation algorithm is also very efficient. From Table 2, we can see that the discretization scheme (10) computed on an adaptive mesh is more accurate and efficient than that computed on the Shishkin mesh. Since the monitor function in [20] is different from the monitor function in this paper, the results obtained by using the method in [20] may be better than our results. However, it is difficult to get the convergence analysis in [20].

In addition, in order to help readers have a deep understanding of adaptive grid method, Figure 1a, which should be read from bottom to top, directly reflects the moving process of the adaptive mesh for $\varepsilon = 10^{-4}$ and $N = 64$. Meanwhile, Figure 1b provides the corresponding graph of numerical solution. Obviously, it is shown that the solution of the test problem has a boundary layer at $x = 0$, which is also clearly reflected in Figure 1.

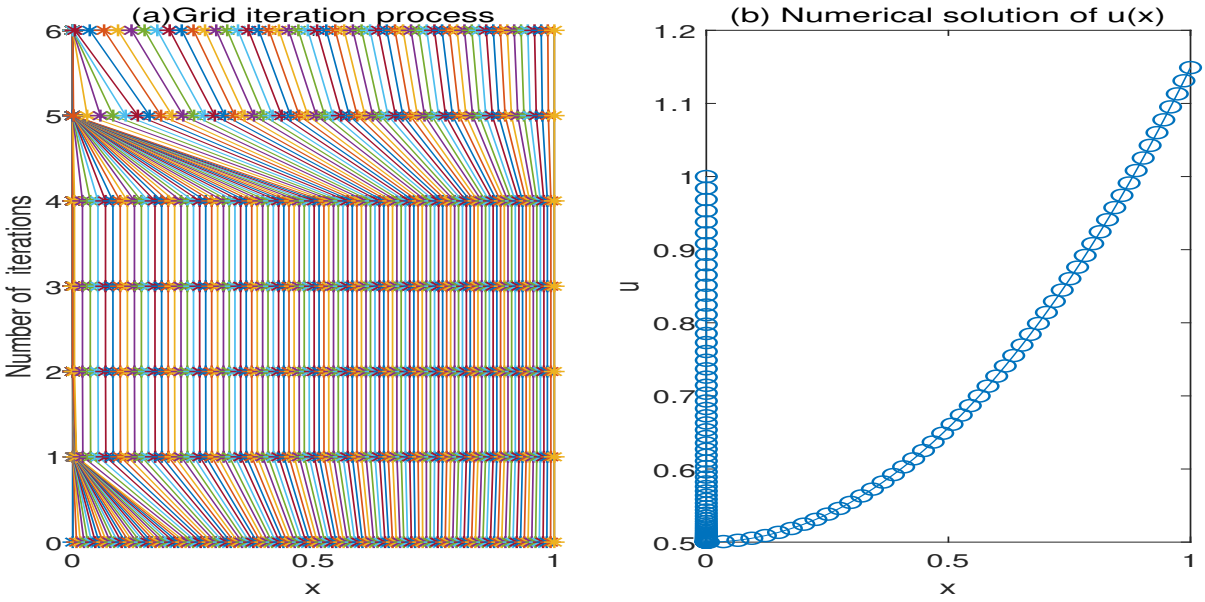

Figure 1. Evolution of the adaptive mesh and numerical solution of Example 1 with $\varepsilon = 10^{-4}$ and $N = 64$. (**a**) Grid iteration process, (**b**) Numerical solution of $u(x)$.

Example 2. The second test example in [1] is

$$\varepsilon u' + u + \int_0^x xu(s)ds = -\frac{\varepsilon}{(1+x^2)} + \frac{1}{1+x} + x\varepsilon\left(1 - e^{-\frac{x}{\varepsilon}}\right) + x\ln(1+x), 0 < x \leq 1,$$
$$u(0) = 2$$

with the analytic solution $u(x) = e^{-\frac{x}{\varepsilon}} + \frac{1}{1+x}$. Then, the maximum point-wise errors are calculated by

$$E_\varepsilon^N = \max_{0 \leq i \leq N} \left| u_i^N - u(x_i) \right|.$$

The rates of convergence are computed by using Equation (39). In order to solve this test Example 2 by using our presented adaptive grid method, we first choose $C_0 = 1.1$ and $\alpha = 1$. Then, Table 3 provides the results obtained using our presented adaptive grid method for $\varepsilon = 10^{-2k}, k = 1, 2, 3, 4$, and $N = 64, 128, 256, 512, 1024, 2048$. In addition, the comparison of numerical results with Shishkin mesh is listed in Table 4. For smaller values of ε, one can see that the convergence rates of the presented adaptive grid are close to 2. For larger values of N, the number of iterations $Iter$ of our adaptive grid generation given in Section 5.1 is also very small. Figure 2 provides the evolution of the above mesh-generation algorithm and the corresponding graph of numerical solution with $\varepsilon = 10^{-4}$, $N = 64$. It is shown that the numerical solution of example 2 has a boundary lay at $x = 0$.

Table 3. Numerical results of our presented adaptive grid method for Example 2.

ε		$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$	$N = 2048$
10^{-2}	E_ε^N	3.99×10^{-4}	1.41×10^{-4}	4.21×10^{-5}	1.10×10^{-5}	2.93×10^{-6}	7.53×10^{-7}
	r_ε^N	1.51	1.75	1.93	1.92	1.96	-
	Iter	2	2	2	1	1	1
10^{-4}	E_ε^N	8.93×10^{-5}	2.23×10^{-5}	5.36×10^{-6}	1.34×10^{-6}	4.15×10^{-7}	1.98×10^{-7}
	r_ε^N	2.01	2.05	2.00	1.69	1.07	-
	Iter	3	3	3	2	2	3
10^{-6}	E_ε^N	8.33×10^{-5}	2.47×10^{-5}	5.37×10^{-6}	1.34×10^{-6}	3.32×10^{-7}	8.33×10^{-8}
	r_ε^N	1.74	2.21	1.99	2.02	1.99	-
	Iter	5	8	4	3	3	3
10^{-8}	E_ε^N	9.87×10^{-5}	2.46×10^{-5}	5.36×10^{-6}	1.33×10^{-6}	3.34×10^{-7}	8.21×10^{-8}
	r_ε^N	2.01	2.20	2.01	1.99	2.02	-
	Iter	7	9	5	5	4	3

Table 4. Comparison of numerical results with the other methods for Example 2.

N	$\varepsilon = 2^{-12}$			$\varepsilon = 2^{-24}$		
	S-Mesh [1]	**Method [20]**	**Our Method**	**S-Mesh [1]**	**Method [20]**	**Our Method**
64	1.03×10^{-2}	0.32×10^{-4}	1.38×10^{-4}	1.07×10^{-2}	4.23×10^{-5}	8.22×10^{-5}
	1.83	2.00	2.03	1.83	2.45	1.91
128	2.89×10^{-3}	0.79×10^{-5}	3.38×10^{-5}	1.02×10^{-3}	0.77×10^{-5}	2.18×10^{-5}
	1.88	1.73	2.10	1.88	2.01	1.84
256	7.84×10^{-4}	2.39×10^{-6}	7.85×10^{-6}	8.21×10^{-4}	1.93×10^{-6}	6.09×10^{-6}
	1.95	1.07	2.16	1.96	2.01	2.20
512	2.03×10^{-4}	1.14×10^{-6}	1.76×10^{-6}	2.11×10^{-4}	0.48×10^{-6}	1.32×10^{-6}
	1.99	1.15	1.42	1.99	2.00	2.01
1024	5.10×10^{-5}	5.12×10^{-7}	6.59×10^{-7}	5.30×10^{-5}	1.19×10^{-7}	3.27×10^{-7}

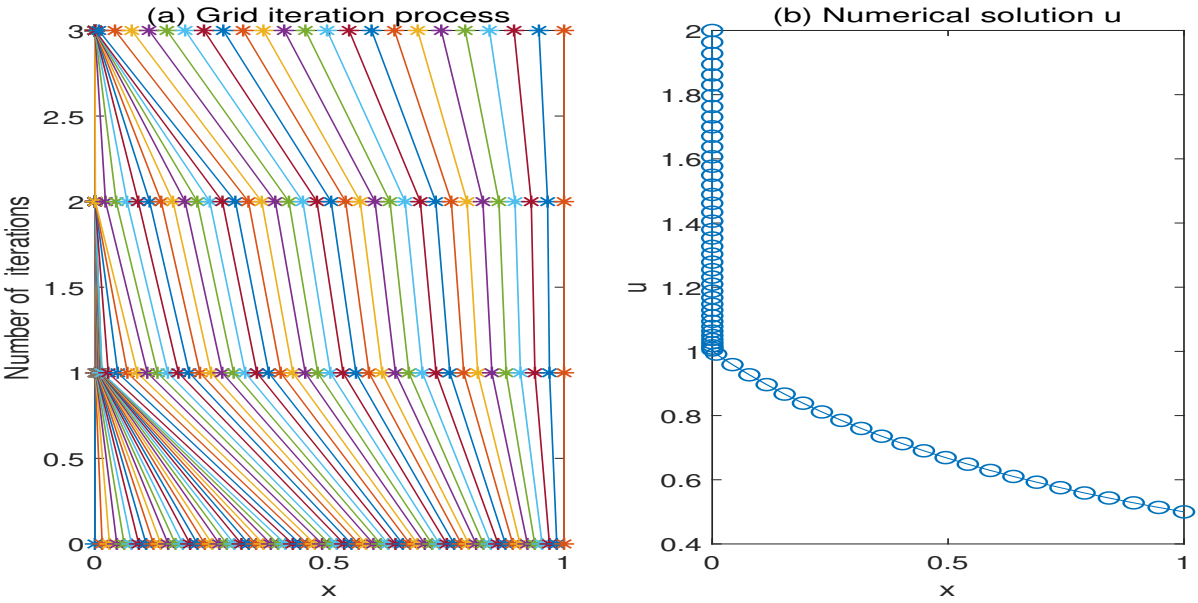

Figure 2. Grid iteration process and numerical solution with $\varepsilon = 10^{-4}$ and $N = 64$ for Example 2. (**a**) Grid iteration process, (**b**) Numerical solution u.

6. Conclusions

As far as we known, most of adaptive grid methods used to solve singularly perturbed problems, which contain a first-order derivative term, are only first-order accurate. For

this reason, based on the fitted finite difference scheme proposed in [1], this paper mainly discussed a second-order adaptive grid method for a singularly perturbed first-order Volterra integrodifferential equation. By using the truncation error analysis of the presented discretization scheme (10), we constructed a suitable monitor function, which is used to design an adaptive grid. It is shown from the convergence analysis that our presented adaptive method is uniformly convergent and independent of the perturbation parameter in the discrete maximum norm.

Author Contributions: Conceptualization, Y.Z.; project administration, L.L.; writing–original draft, Y.L.; and writing–review editing, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Natural Science Foundation of Guangxi province (2020GXNSFAA159010), the Open Research Fund of Guangxi Key Lab of Human-machine Interaction and Intelligent Decision (GXHIID2209) and the projects of Excellent Young Talents Fund in Universities of Anhui Province(gxyq2021225).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yapman, M.; Amiraliyev, G.M. A novel second-order fitted computational method for a singularly perturbed Volterra integro-differential equation. *Int. J. Comput. Math.* **2020**, *97*, 1293–1302. [CrossRef]
2. Brunner, H. *Collocation Methods for Volterra Integral and Related Functional Equations*; Cambridge University Press: Cambridge, UK, 2004.
3. Rudenko, O.V. Nonlinear integro-differential models for intense waves in media like biological tissues and geostructures with complex internal relaxation-type dynamics. *Acouset. Phys.* **2014**, *60*, 398–404. [CrossRef]
4. Bouchra, A. Qualitative analysis and simulation of a nonlinear integro-differential system modelling tumor-immune cells competition. *Int. J. Biomath.* **2018**, *11*, 1850104.
5. de Gaetano, A.; Arino, O. Mathematical modelling of the intravenous glucose tolerance test. *J. Math. Biol.* **2000**, *40*, 136–168. [CrossRef] [PubMed]
6. Abdul, I. *Introduction to Integral Equations with Application*; Wiley: New York, NY, USA, 1999.
7. Amiraliyev, G.M.; Şevgin, S. Uniform difference method for singularly perturbed Volterra integro-differential equations. *Appl. Math. Comput.* **2005**, *179*, 731–741. [CrossRef]
8. Salama, A.A.; Bakr, S.A. Difference schemes of exponential type for singularly perturbed Volterra integro-differential problems. *Appl. Math. Model.* **2007**, *3*131, 866–879. [CrossRef]
9. Ramos, J.I. Exponential techniques and implicit Runge-Kutta method for singularly-perturbed Volterra integro-differential equations. *Neural Parallel.* **2008**, *16*, 387–404.
10. Şevgin, S. Numerical solution of a singularly perturbed Volterra integro-differential equation. *Adv. Differ. Equations* **2014**, *2014)*, 171–196. [CrossRef]
11. Iragi, B.C.; Munyakazi, J.B. A uniformly convergent numerical method for a singularly perturbed Volterra integro-differential equation. *Int. J. Comput. Math.* **2020**, *97*, 759–771. [CrossRef]
12. Huang, J.; Cen, Z.; Xu, A.; Liu, L.-B. A posteriori error estimation for a singularly perturbed Volterra integro-differential equation. *Numer. Algorthms* **2020**, *83*, 549–563. [CrossRef]
13. Kauthen, J.P. Implicit Runge-Kutta methods for some singularly perturbed Volterra integro-differential-algebraic equation. *Appl. Numer. Math.* **1993**, *13*, 125–134. [CrossRef]
14. Kauthen, J.P. Implicit Runge-Kutta methods for singularly perturbed integro-differential systems. *Appl. Numer. Math.* **1995**, *18*, 201–210. [CrossRef]
15. Kopteva, N.; Stynes, M. A robust adaptive method for a quasilinear one-dimensional convection-diffusion problem. *SIAM J. Numer. Anal.* **2001**, *39*, 1446–1467. [CrossRef]
16. Linß, T. Analysis of a system of singularly perturbed convection-diffusion equations with strong coupling. *SIAM J. Numer. Anal.* **2009**, *47*, 1847–1862. [CrossRef]
17. Roos, H.-G.; Stynes, M.; Tobiska, L. *Robust Methods for Singularly Perturbed Differential Equations*, 2nd ed.; Springer Series in Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 2008; Volume 24.
18. Kumar, S.S.; Vigo-Aguiar, J. Analysis of a nonlinear singularly perturbed Volterra integro-differential equation. *J. Comput. Appl. Math.* **2021**, *404*, 113410.

19. Long, G.; Liu, L.-B.; Huang, Z. Richardson extrapolation method on an adaptive grid for singularly perturbed Volterra integro-differential equations. *Numer. Funct. Anal. Optim.* **2021**, *42*, 739–757. [CrossRef]
20. Luo, X.; Yang, N.; Tong, Q. A Novel Second-Order Adaptive Grid Method for Singularly Perturbed Convection-Diffusion Equations. *J. Uncertain Syst.* **2021**, *14*, 2150026. [CrossRef]
21. Amiraliyev, G.M.; Yilmaz, B. Fitted difference method for a singularly perturbed initial value problem. *Int. J. Math. Comput.* **2014**, *22*, 1–10.
22. Kudu, M.; Amirali, I.; Amiraliyev, G.M. A finite-difference method for a singularly perturbed delay integro-differential equation. *J. Comput. Appl. Math.***2016**, *308*, 379–390. [CrossRef]
23. Mackenzie, J. Uniform convergence analysis of an upwind finite-difference approximation of a convection-diffusion boundary value problem on an adaptive grid. *IMA J. Numer. Anal.* **1999**, *19*, 233–249. [CrossRef]
24. Beckett, G.; Mackenzie, J.A. Convergence analysis of finite difference approximations to a singularly perturbed boundary value problem. *Appl. Numer. Math.* **2000**, *35)*35:87–109. [CrossRef]
25. Kopteva, N.; Madden, N.; Stynes, M. Grid equidistribution for reaction-diffusion problems in one dimension. *Numer. Algorthms* **2005**, *40*, 305–322. [CrossRef]
26. Qiu, Y.; Sloan, D.M.; Tang, T. Numerical solution of a singularly perturbed two point boundary value problem using equidistribution: analysis of convergence. *J. Comput. Appl. Math.* **2000**, *116*, 121–143. [CrossRef]
27. Chen, Y. Uniform pointwise convergence for a singularly perturbed problem using arc-length equidistribution. *J. Comput. Appl. Math.***2003**, *159*, 25–34. [CrossRef]