SEMI-LINEAR DIFFERENCE SCHEMES*

SUN JIA-CHANG[1)] (孙家昶)
(*Computing Center, Academia Sinica, Beijing, China*)

Abstract

A class of semi-linear numerical differentiation formulas is designed for functions with steep gradients. A semi-linear second-order difference scheme is constructed to solve the two-point singular perturbation problem $-\epsilon u''+p(x)u'+q(x)u=f(x)$, $u(0)=u(1)=0$. It is shown that this semi-linear scheme has one more order of approximation precision than the central difference scheme for small ϵ and saves computation time for required accuracy. Numerical results agreeing with the above analysis are included.

1. Introduction

Numerical differential formulas play a very important role in constructing difference schemes of differential equations. However, the usual numerical differentiation formulas based on polynomial approximations may lead to very poor results when the functions are not smooth. Usually there are two ways to avoid this difficulty; one is to refine the mesh, and the other is to use a higher order polynomial interpolation.

A different approach is introduced in this paper by considering "weak" nonlinear numerical differentiation formulas beyond linear functional approximation. In section 2, we derive some semi-linear numerical differentiation formulas. Such a scheme is semi-linear as an operator; besides, the numerical differentiation formula, chosen for a function with steep gradients, should depend on the behavior of the function. As an example, a detailed analysis is given in section 3 for a model problem: $-\epsilon u''+u'=0$, $u(0)=0$, $u(1)=1$, which was recently discussed by other authors[1]-[3]. In section 4, we consider a more general elliptic singular perturbation problem: $-\epsilon u''+pu'+qu=f$. The semi-linear scheme presented there is shown to have one more order of precision than the conventional central difference scheme for the singular perturbation problem if $h\leqslant 2\epsilon/\|p\|_\infty$, where h is a uniform mesh size in the "singular" subdomain. In the larger "regular" subdomain the mesh size can be used as large as one desires. While maintaining the same accuracy, the semi-linear scheme costs less CPU time than the linear scheme. There is a simple way to reduce the resulting semi-linear system to an iteration with the corresponding linear system. The numerical tests presented in section 5 match the above analysis very well. A similar study in the two-dimensional case will appear in another separate paper.

2. Semi-linear Numerical Differentiation Formulas

Let $u(x)$ be a function defined in (a, b) with large derivatives. Without loss of

* Received September 9, 1982.

[1)] This research was supported in part by ONR Grant N00014-82-K-0184, and Yale University when the author was visiting the Dept. of Computer Science of Yale.

generality, suppose $u(x)$ is monotonic in the interval, and $u=Fx$ is a one-to-one map.

Denote $x_{-1}=a<x_0<x_1=b$, and let G be defined as an indefinite integral of F such that

$$G(x)=\int Fx\,dx. \tag{1}$$

By the mean value theorem, there exist two points z_{-1} and z_1: $x_{-1}<z_{-1}<x_0<z_1<x_1$, such that

$$[F^{-1}u_0,\ F^{-1}u_{-1}]G=u(z_{-1}),\quad [F^{-1}u_1,\ F^{-1}u_0]G=u(z_1), \tag{2}$$

where $[x_1,\ x_2]Y$ denotes the divided difference: $[x_1,\ x_2]Y=\dfrac{Y(x_1)-Y(x_2)}{x_1-x_2}$. Now we look for an approximate formula for the first derivative at the node $x=x_0$, based on the formulas (2), as follows:

$$u'(x_0)\sim\frac{2}{x_1-x_{-1}}([F^{-1}u_1,\ F^{-1}u_0]G-[F^{-1}u_0,\ F^{-1}u_{-1}]G). \tag{3}$$

If we take F as the identity map, and $G(x)=x^2/2$, then (3) is just the usual central difference formula based on the quadratic interpolation. In general, we assume F to be an admissible one-to-one map such that G can be obtained from (1) directly. For any such F, (3) defines a numerical formula for the first derivative at the node $x=x_0$. As an example, let $F^{-1}f=f^r$, where r is a real parameter. Suppose $u(x)>0$ in $(x_{-1},\ x_1)$; from (3) and (1), we obtain

$$u'(x_0)\doteq\frac{2r}{(1+r)(x_1-x_{-1})}\left\{\frac{u_1^{1+r}-u_0^{1+r}}{u_1^r-u_0^r}-\frac{u_0^{1+r}-u_{-1}^{1+r}}{u_0^r-u_{-1}^r}\right\}, \tag{4}$$

where $u_j=u(x_j)$, $j=-1,\ 0,\ 1$.

When $r=\dfrac{1}{2}$, $x_0=\dfrac{1}{2}(x_1+x_{-1})$, (4) becomes

$$u'(x_0)=\frac{1}{3h}(u_1^{1/2}-u_{-1}^{1/2})(u_1^{1/2}+u_0^{1/2}+u_{-1}^{1/2}),\qquad h=x_1-x_0.$$

Theorem 1. *Let $u, F\in C^k(x_{-1},\ x_1)$ where $k=3$ or 4, $F^{-1}u$ is a one-to-one map, $h=x_0-x_{-1}=x_1-x_0$. Then, the remainder of the numerical differentiation formula* (3) *equals*

$$\begin{aligned}&\frac{1}{h}([F^{-1}u_1,\ F^{-1}u_0]G-[F^{-1}u_0,\ F^{-1}u_{-1}]G)-u'(x_0)\\ &\quad=\frac{h^2}{12}\frac{d}{dx}\left\{2u''+u'^2\frac{d^2F^{-1}u}{du^2}\left(\frac{dF^{-1}u}{du}\right)^{-1}\right\}\Bigg|_{x=x_0}+O(h^k).\end{aligned} \tag{5}$$

Proof. Applying the Taylor expansion for $G(y)$ upon z, one obtains

$$\begin{aligned}G(y)-G(z)=&(y-z)G'(z)+(y-z)^2G''(z)/2+(y-z)^3G^{(3)}(z)/3!\\&+(y-z)^4G^{(4)}(z)/4!+O((y-z)^5).\end{aligned}$$

Hence

$$\begin{aligned}W(y_1,\ y_0,\ y_{-1})&\equiv\frac{G(y_1)-G(y_0)}{y_1-y_0}-\frac{G(y_0)-G(y_{-1})}{y_0-y_{-1}}\\&=\frac{y_1-y_{-1}}{2}G''(y_0)+\frac{(y_1-y_0)^3+(y_0-y_{-1})^3}{4!}G^{(4)}(y_0)+O((y_1-y_{-1})^5).\end{aligned}$$

By means of rules for finding the derivative function in the implicit case, it is easily seen that

$$W(F^{-1}u_1,\ F^{-1}u_0,\ F^{-1}u_{-1})$$
$$=hu'(x_0)+\frac{h^3}{12}\,\frac{d}{dx}\left\{2u''+u'^2\,\frac{d^2F^{-1}u}{du^2}\left(\frac{dF^{-1}u}{du}\right)^{-1}\right\}\Bigg|_{x=x_0}+O(h^{k+1}).$$

Corollary 1. If the map F satisfies the relation

$$\frac{d}{dx}\left\{2u''+u'^2\,\frac{d^2F^{-1}u}{du^2}\left(\frac{dF^{-1}u}{du}\right)^{-1}\right\}\Bigg|_{x=x_0}=0,$$

then formula (3) has an error of third or fourth order for $k=3$ or 4, respectively.

Corollary 2. If $k=2$, the error estimate is of second order, and the second term on the right side of (5) is evaluated at a point $\xi\in(x_{-1},\ x_1)$.

The dominant term of the truncation error of formula (4) is

$$\frac{h^2}{12}\,\frac{d}{dx}\left\{2u''+(r-1)\frac{u'^2}{u}\right\}\Bigg|_{x=x_0}.$$

A fourth order numerical differentiation formula may be obtained if r is so chosen that the above term vanishes; for instance,

$$r=1-2\,\frac{uu''}{u'^2}\Bigg|_{x=x_0}\quad\text{or}\quad r=-1+2\left(\frac{u_1}{u_1'}-\frac{u_{-1}}{u_{-1}'}\right).$$

Both schemes above are implicit. However, they are useful in constructing a high order difference scheme for differential equations, even for partial differential equations.

The limiting case of (4) in which r tends to zero is interesting for solving singular perturbation problems

$$u'(x_0)=\frac{1}{h}\left\{\frac{u(x_1)-u(x_0)}{\log(u(x_1)/u(x_0))}-\frac{u(x_0)-u(x_{-1})}{\log(u(x_0)/u(x_{-1}))}\right\}$$
$$-\frac{h^2}{12}\,\frac{d}{dx}\left\{2u''-\frac{u'^2}{u}\right\}\Bigg|_{x=x_0}+O(h^4). \tag{6}$$

From Theorem 1 it follows that formula (4) is of fourth order if the function u satisfies

$$2uu''-(1-r)u'^2=bu,$$

where $b=$constant. One particular solution for $b=0$ is a power function

$$u(x)=(cx+d)^{\frac{2}{1-r}},$$

where c, d are constants. This means that the "logarithmic type" numerical differentiation formula (6), specifically designed for solutions with large derivatives, is also good for smooth functions.

Similarly, we may derive some numerical differentiation formulas for higher derivatives. For instance, there is an analogue of Theorem 1 for second derivatives.

Theorem 2. *If u, $F\in C^4(x_{-1},\ x_1)$, $h=x_0-x_{-1}=x_1-x_0$, and $F^{-1}u$ is a one-to-one map, then*

$$u''(x_0)=\frac{2}{h^2}\{[F^{-1}u_1,\ F^{-1}u_0]G+[F^{-1}u_0,\ F^{-1}u_{-1}]G-2u_0\}$$
$$-\frac{u_0'^2}{3}\,\frac{d^2F^{-1}u}{du^2}\left(\frac{dF^{-1}u}{du}\right)^{-1}_{x_0}+O(h^2), \tag{7}$$

where G is defined by (1) *and u_0' is given by* (3).

The proof of Theorem 2 follows from the proof of Theorem 1, using the Taylor expansion and differentiation of implicit functions. (7) is an extension of the second order central difference scheme; the latter is only a particular case of $F=I$.

3. Model Problem Analysis

In this section our attention is devoted to the following model problem, discussed by various authors, cf. Barrett and Morton[1], Christie *et al.* [2]–[3],

$$Lu=-\epsilon u''+u'=0 \quad \text{in} \quad (0,\,1),$$

$$u(0)=0, \quad u(1)=1, \tag{8}$$

with solution

$$u(x)=\frac{e^{x/\epsilon}-1}{e^{1/\epsilon}-1}. \tag{9}$$

Let $x_j=jh(j=0,\,1,\,\cdots,\,N;\,h=1/N)$. Using the conventional central difference for the first and second derivatives in (8) leads to the following difference equation

$$L_hU_j^h=-\left(a+\frac{1}{2}\right)U_{j-1}^h+2aU_j^h+\left(\frac{1}{2}-a\right)U_{j+1}^h=0, \quad 1\leqslant j\leqslant N-1, \tag{10}$$

with $U_0^h=0$ and $U_N^h=1$, where $a\equiv\frac{\epsilon}{h}$.

The exact solution of the difference equation (10) is given by

$$U_j^h=\frac{p(h)^j-1}{p(h)^N-1},$$

where

$$p\equiv\frac{a+1/2}{a-1/2}. \tag{11}$$

In order to preserve the increasing monotonic property of the solution (9), it is reasonable to demand

$$a>\frac{1}{2}, \quad \text{i.e.} \quad h<2\epsilon. \tag{12}$$

Let $E_j^h=u(x_j)-U_j^h$, $E^h=\max E_j^h$. From (9), (11) and (12), for small ϵ, we have an asymptotic estimate for small h,

$$0<E_j^h<E_{N-1}^h=E^h\sim\frac{1}{12}\,a^{-2}=\frac{h^2}{12\epsilon^2}. \tag{13}$$

Hence, the difference scheme doesn't converge for small ϵ, if the mesh size h is of the same order of ϵ. Meanwhile, (13) shows that the maximum error always occurs at the last interior mesh point. This shows that the scheme (10) yields a poor approximate solution near the layer boundary at the right-hand end point. In fact, substituting the exact solution (9) into (10) and using equation (8) yields the local truncation error

$$\begin{aligned}\mathrm{Tr}\;u_j&\equiv-\frac{\epsilon}{h}(u_{j+1}-2u_j+u_{j-1})+\frac{1}{2}(u_{j+1}-u_{j-1})\\&=\frac{h^3}{12}\,u_j^{(3)}+O(h^5)=\frac{1}{12}\;\frac{e^{x/\epsilon}}{(e^{1/\epsilon}-1)a^3}+O(h^5).\end{aligned}$$

From (9), note that when ϵ tends to 0, both terms $u^{(3)}(x)$ and $\left\{\frac{u'^2}{u}\right\}'$ tend to infinity as x near 1, but their difference

$$u^{(3)}(x)-\frac{d}{dx}\,\frac{u'^2}{u}\to\epsilon^{-3}e^{-1/\epsilon}\to0.$$

Hence, we suggest adopting the "logarithmic type" numerical differential formula (6) near the layer boundary. Thus, the difference scheme (10) is revised as follows:

$$L_hU_j^h=-\left(a+\frac{1}{2}\right)U_{j-1}^h+2aU_j^h+\left(\frac{1}{2}-a\right)U_{j+1}^h,\quad \text{for}\quad 2j<N,$$

$$\begin{aligned}L_hU_j^h=&-\left(a+\frac{1}{2}\right)U_{j-1}^h+2aU_j^h+\left(\frac{1}{2}-a\right)U_{j+1}^h\\&-Q(U_{j-1}^h,\ U_j^h,\ U_{j+1}^h),\quad \text{for}\quad N\leqslant 2j<2N,\end{aligned}\tag{14}$$

where

$$Q(U_{j-1},\ U_j,\ U_{j+1})=\frac{U_{j+1}-U_{j-1}}{2}-\left\{\frac{U_{j+1}-U_j}{\log(U_{j+1}/U_j)}-\frac{U_j-U_{j-1}}{\log(U_j/U_{j-1})}\right\},\tag{15}$$

with $U_0^h=0$ and $U_N^h=1$, $a=\frac{\epsilon}{h}$.

The local truncation error of the scheme becomes

$$L_hu_j=\begin{cases}\dfrac{h^3}{12}u_j^{(3)}, & \text{if}\quad 2j<N,\\[2ex] \dfrac{h^3}{12}\left\{u^{(3)}-\dfrac{d}{dx}\ \dfrac{u'^2}{u}\right\}\Big|_j, & \text{if}\quad N\leqslant 2j<2N,\end{cases}$$

or

$$12h^{-3}\,\mathrm{Tr}\,u_j^h=\begin{cases}\dfrac{e^{x/\epsilon}}{(e^{1/\epsilon}-1)\epsilon^3}, & \text{if}\quad 2j<N,\\[2ex] \dfrac{e^{x/\epsilon}}{(e^{x/\epsilon}-1)^2\epsilon^3}, & \text{if}\quad N\leqslant 2j<2N,\end{cases}\tag{16}$$

where $x=jh$. The maximum local truncated error of (14) can be easily found to be

$$\mathrm{Tr}\,u^h\equiv\max\,\mathrm{Tr}\,u_j^h\sim\frac{h^3}{12}\ \frac{e^{1/(2\epsilon)}}{\epsilon^3(e^{1/(2\epsilon)}-1)^2}\sim\frac{h^3}{12}\ e^{-1/(2\epsilon)}\epsilon^{-3}.\tag{17}$$

Note that there is a striking contrast between the truncation errors for the revised scheme (14) and for the original linear scheme (10). The maximum truncation error has no power of the mesh size h for (10), but is of second order for (14) with a coefficient which tends to 0 as ϵ does.

Now we discuss how to solve the resulting system

$$AU=d+Q(U),\tag{18}$$

where A and d coincide with the corresponding linear scheme, and the non-linear term $Q(U)$ arises from using the semi-linear scheme for $x>\frac{1}{2}$.

Let $D_n=\det(A_n)$, $\beta_n=\frac{D_{n-1}}{D_n}$, where

$$A_n=\begin{vmatrix}2a & -(a-1/2) & & & \\ -(a+1/2) & 2a & -(a-1/2) & & \\ & \cdots\cdots\cdots & \cdots\cdots\cdots & \cdots\cdots & \\ & & \cdots\cdots\cdots & \cdots\cdots\cdots & \cdots\cdots \\ & & -(a+1/2) & 2a & -(a-1/2)\\ & & & -(a+1/2) & 2a\end{vmatrix},$$

then, we have the following recursions

$$D_n=2aD_{n-1}-\left(a^2-\frac{1}{4}\right)D_{n-2},\qquad D_0=1,\quad D_1=2a,$$

$$\beta_n=\frac{1}{2a-(a^2-1/4)\beta_{n-1}}, \qquad \beta_{-1}\equiv 0.$$

For $a\geqslant\frac{1}{2}$, $\beta\equiv\lim\beta_n=\frac{1}{a+1/2}$, and

$$0\leqslant\beta_{n+1}-\beta_n\leqslant\frac{a-1/2}{a+1/2}(\beta_n-\beta_{n-1}), \qquad \frac{1}{2a}\leqslant\beta_n\leqslant\beta\equiv\lim\beta_n=\frac{1}{a+1/2}.$$

The inverse matrix A^{-1} can be easily found such that

$$A^{-1}=(\alpha_{i,j}^{-1}),$$

where

$$\alpha_{i,j}^{-1}=\begin{cases}(a+1/2)^{i-j}D_{j-1}D_{n-i}/D_n, & \text{if} \quad i\geqslant j,\\ (a-1/2)^{j-i}D_{i-1}D_{n-j}/D_n, & \text{if} \quad i\leqslant j.\end{cases} \tag{19}$$

Hence, we have

Lemma 3. *For $a\geqslant\frac{1}{2}$, A^{-1} is a positive matrix given by* (19). *Moreover,*

$$\alpha_{i,j}^{-1}\geqslant\alpha_{i,j-1}^{-1}, \quad \text{if} \quad i\geqslant j, \qquad \alpha_{i,j}^{-1}\leqslant\alpha_{i,j-1}^{-1}, \quad \text{if} \quad i<j, \tag{20}$$

$$\|A^{-1}\|_\infty\leqslant(N-1)\beta_{N-1}<\frac{1}{h(a+1/2)}.$$

We need the following Lemmas to estimate the nonlinear influence of Q in (18).

Lemma 4. *If a, $b>0$, then*

$$(ab)^{1/2}\leqslant\frac{a-b}{\log(a/b)}\leqslant\frac{a+b}{2}, \tag{21}$$

where the equality holds if and only if $a=b$.

Proof. In fact, integrating t from 0 to 1 on both sides of the inequalities $a^{1-t}b^t\leqslant(1-t)a+tb$ gives the right hand side of (21), and the left-hand side of (21) can be obtained by

$$\int_0^1\left(\frac{a}{b}\right)^{t-1/2}dt=\int_0^{1/2}\left\{\left(\frac{a}{b}\right)^{t-1/2}+\left(\frac{b}{a}\right)^{t-1/2}\right\}dt\geqslant 1.$$

Corollary. For a, $b>0$,

$$0\leqslant r(a,\ b)\equiv\frac{a+b}{2}-\frac{a-b}{\log(a/b)}\leqslant\frac{1}{2}(b^{1/2}-a^{1/2})^2. \tag{22}$$

A straightforward computation yields

Lemma 5. *If a, b and c are positive and $Q(a,\ b,\ c)\equiv r(b,\ c)-r(a,\ b)$, then*

(i) $-\frac{1}{2}(b^{1/2}-a^{1/2})^2\leqslant Q(a,\ b,\ c)\leqslant\frac{1}{2}(c^{1/2}-b^{1/2})^2$.

(ii) *If $c\geqslant b\geqslant a>0$ and $b^2\leqslant ac$, then $Q(a,\ b,\ c)\geqslant 0$ with "$=$" iff $c=a$.*

Now we estimate $\|A^{-1}J(Q(U))\|$, where $J(Q)$ is the Jacobi matrix of Q. Note that each component Q_i in (15) is homogeneous with respect to the variables. Hence, due to the well-known Euler theorem we have the following relationship

$$\{J(Q(u))u\}_j=\{Q(u)\}_j, \quad \text{if} \quad j<N-1, \tag{23}$$

which greatly simplifies the estimations of $\|A^{-1}J(Q(U))\|$. Hence

$$J(Q(u))u=\{0,\ \cdots,\ 0,\ Q_n,\ \cdots,\ Q_{N-2},\ Q_{N-1}^*\},$$

where

$$Q_j = r_{j+1/2} - r_{j-1/2}, \quad r_{j-1/2} \equiv \frac{u_j + u_{j-1}}{2} - \frac{u_j - u_{j-1}}{\log(u_j/u_{j-1})}, \quad j = n, \cdots, N-2,\ 2n = N,$$

$$Q^*_{N-1} = Q_{N-1} - \frac{\partial Q_{N-1}}{\partial u_N} u_N.$$

Suppose $1 > u_j > 0$, $1 < j < N$, $u_N \equiv 1$ for $n \leqslant i \leqslant N-1$. A straightforward computation yields

$$\sigma_i \equiv \{A^{-1} J(Q(u)) u\}_i = \alpha^{-1}_{i,N-1} R_{N,N-1} - \sum_{j=n+1}^{N-1} \{\alpha^{-1}_{i,j} - \alpha^{-1}_{i,j-1}\} r_{j-1/2} - \alpha^{-1}_{i,n} r_{n-1/2}$$

$$= \alpha^{-1}_{i,N-1} R_{N,N-1} + \sum_{j=i+1}^{N-1} \{\alpha^{-1}_{i,j-1} - \alpha^{-1}_{i,j}\} r_{j-1/2} - \sum_{j=n+1}^{i} \{\alpha^{-1}_{i,j} - \alpha^{-1}_{i,j-1}\} r_{j-1/2} - \alpha^{-1}_{i,n} r_{n-1/2},$$

where $$R_{N,N-1} \equiv r_{N-1/2} - \frac{\partial r_{N-1/2}}{\partial u_N} = \frac{u_{N-1}}{2} - \frac{u_{N-1}}{\log u_{N-1}} - \frac{1 - u_{N-1}}{(\log u_{N-1})^2}.$$

Using Lemma 4, for $0 < u < 1$, we have

$$\frac{u}{2} - \frac{u}{\log u} - \frac{1-u}{(\log u)^2} \leqslant \frac{u}{2} + \frac{u}{\log 1/u} - \frac{\sqrt{u}}{\log 1/u}$$

$$\leqslant \sqrt{u} \left\{ \frac{\sqrt{u}}{2} - \frac{\sqrt{u} - 1}{2 \log u^{1/2}} \right\} \leqslant \frac{\sqrt{u}}{2} \{\sqrt{u} - u^{1/4}\}$$

and

$$\frac{u}{2} - \frac{u}{\log u} - \frac{1-u}{(\log u)^2} \geqslant \frac{u}{2} - \frac{u}{\log u} - \frac{1+u}{2 \log 1/u} = \frac{u}{2} - \frac{u-1}{2 \log u} \geqslant \frac{u}{2} - \frac{1+u}{4}.$$

Hence, $$-\frac{1}{4} \leqslant -\frac{1}{4}(1 - u_{N-1}) \leqslant R_{N,N-1} \leqslant -\frac{u^{3/4}_{N-1}}{2}(1 - u^{1/4}_{N-1}) < 0.$$

Let $r_M \equiv \max r_{j-1/2}$. Using (22) and (20) leads to

$$\alpha^{-1}_{i,N-1} R_{N,N-1} - r_M \alpha^{-1}_{i,i} \leqslant \sigma_i \leqslant \alpha^{-1}_{i,N-1} R_{N,N-1} + r_M (\alpha^{-1}_{i,i} - \alpha^{-1}_{i,N-1}). \tag{24}$$

From (19), $\left(a + \frac{1}{2}\right)^{-1} \geqslant \alpha^{-1}_{i,i} \geqslant \alpha^{-1}_{i,N-1}$. Hence, inequality (24) implies

$$-\left(r_M + \frac{1}{4}\right) \leqslant \left(a + \frac{1}{2}\right) \sigma_i < r_M.$$

From (22), $0 \leqslant r_M \leqslant \frac{1}{2}$ if $0 < u_j \leqslant 1$. Finally, we obtain an upper bound

$$\|A^{-1} J(Q(u))\| = \sup_{\|u\|=1} \|A^{-1} J(Q(u)) u\| \leqslant \frac{3}{4(a + 1/2)}. \tag{25}$$

As a consequence, the following two theorems are obtained:

Theorem 6. *The map*

$$P(u) \equiv A^{-1} Q(u)$$

is contractive and has a unique fixed point if

$$a = \frac{\epsilon}{h} \geqslant \frac{1}{2}. \tag{26}$$

Theorem 7. *When $h \leqslant 2\epsilon$, the semi-linear scheme* (14) *has a unique solution, and it can be solved by the following "simple" iteration*

$$AU^{(0)} = d, \qquad AU^{(k)} = d + Q(U^{(k-1)}), \qquad k > 1. \tag{27}$$

Besides, from (11) and Lemma 5, U^0_j is a monotonic increasing sequence of j and $Q(U^0)$ is nonnegative. By induction, it can be proved that, for the model problem (8), U^k_j in the iteration (27) preserve the monotonically increasing property with j

for each k. Hence, when k tends to ∞, the monotonically increasing property still remains true, i.e.,

$$U_j \leqslant U_{j+1}, \qquad j=0, 1, 2, \cdots.$$

Now we consider the convergence in another meaning: the convergence of the solution U^h of the nonlinear difference equation (14) to the exact solution u of the differential equation (8). Substituting the exact solution u into the scheme (18), we have

$$Au=d+Q(u)+\mathrm{Tr}(u),$$

where $\mathrm{Tr}(u)$ is the truncation error vector. Subtracting (18) from the above yields the error equation

$$A(u-U^h)=Q(u)-Q(U^h)+\mathrm{Tr}(u).$$

When $h<2\epsilon$ and A^{-1} exists, the above formula is equivalent to

$$u-U^h=A^{-1}(Q(u)-Q(U^h))+A^{-1}\mathrm{Tr}(u). \tag{28}$$

According to Lemma 6, for $h<2\epsilon$, $A^{-1}Q(u)$ is a contraction map in the maximum norm. Hence, from (25),

$$\|A^{-1}(Q(u)-Q(U^h))\|_\infty \leqslant \frac{3}{4(a+1/2)}\|u-U^h\|_\infty.$$

Applying Lemma 3 and (17), we obtain

$$\|u-U^h\|_\infty < \left\{1-\frac{3}{4(a+1/2)}\right\}^{-1}\|A^{-1}\|_\infty \|\mathrm{Tr}(u)\|_\infty < Ch^2,$$

where the constant

$$C=\frac{1}{3}e^{-1/(2\epsilon)}\epsilon^{-3} \leqslant \frac{1}{3}\left\{\frac{6}{e}\right\}^3 < 3.6. \tag{29}$$

Therefore, we have proved the following main result of this section:

Theorem 8. *The solution of the semi-linear scheme* (14) *converges to the exact solution of the singular perturbation problem* (8) *with second order rate in the following sense*

$$\|U^h-u\|_\infty < Ch^2, \qquad \text{if} \quad h \leqslant 2\epsilon, \tag{30}$$

where the coefficient C defined in (29) *is uniformly bounded for all ϵ and tends to zero as ϵ does.*

Note that the restricted mesh condition $h<2\epsilon$, caused by introducing the semi-linear scheme, is only needed in the steeper gradient interval $x>1/2$. Hence, it is possible to restrict the mesh condition only in the interval. As a matter of fact, in block matrix form, the scheme (14) can be written as

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} V \\ U \end{pmatrix}=\begin{pmatrix} 0 \\ d \end{pmatrix}+\begin{pmatrix} 0 \\ Q(U, V) \end{pmatrix}.$$

Let h_1 and h be two different uniform mesh sizes used for $x<1/2$ and $x \geqslant 1/2$, respectively, and let $a=\frac{\epsilon}{h}$, $a_1=\frac{\epsilon}{h_1}$.

Applying Gaussian elimination of matrix form, the above system can be reduced to

$$\tilde{A}_{22}U=d+Q(U, -A_{11}^{-1}A_{12}U), \tag{31}$$

where $$\widetilde{A}_{22}=(\alpha^*_{ij}),\quad \alpha^*_{11}=a-\frac{1}{2}+\frac{a+1/2}{a_1+1/2},\ \alpha^*_{ij}=\alpha_{ij}\quad \text{if}\quad (i,j)\neq(1,1).$$

When $a\geqslant a_1$, i.e., $h\leqslant h_1$, Lemma 3 still holds. Hence, the above conclusion can be improved further by the next assertion.

Theorem 9. *If*

$$h\leqslant\min(h_1, 2\epsilon), \tag{32}$$

then the iterative procedure (27) *converges for the system* (31) *and the error estimate* (30) *is still valid.*

4. Linear Second Order Two-point Boundary Layer Problems

Now we consider a more general linear second order singular perturbation problem

$$\begin{gathered} Lu\equiv-\epsilon u''+p(x)u'+q(x)u=f(x),\quad \text{in}\quad [0,1], \\ u(0)=u(1)=0, \end{gathered} \tag{33}$$

where ϵ is a small positive parameter and $p(x), q(x)$ and $f(x)$ are sufficiently smooth that their second derivatives are uniformly bounded for all x in $[0, 1]$ and for all $\epsilon>0$; besides, $p(x)\geqslant p^*>0$, $q(x)\geqslant 0$ and $q(x)-p'(x)/2\geqslant\delta>0$ on $[0, 1]$. Let

$$L_hU^h_j\equiv-(\alpha+p_j/2)U^h_{j-1}+(2\alpha+q_jh^2)U^h_j+(p_j/2-\alpha)U^h_{j-1}. \tag{34}$$

The interval $[0, 1]$ is divided into two subintervals: $[0, 1]=I_r+I_s$, where I_r is called a regular subinterval over which the first derivative of $u(x)$ is bounded by a control number μ^h, and I_s defines a singular subinterval over which $u'(x)$ may be very large.

Similar to (14), in this case the semi-linear scheme becomes

$$L_hU^h=\begin{cases} hf^h_j, & \text{if}\quad \left|p_j\dfrac{U_{j+1}-U_{j-1}}{2h}\right|\leqslant\mu^h, \\ hf^h_jhf^h_j-g_j(U^h_{j-1}, U^h_j, U^h_{j+1}) & \text{if}\quad \left|p_j\dfrac{U_{j+1}-U_{j-1}}{2h}\right|>\mu^h; \end{cases} \tag{35}$$

where $$g_j(u_{j+1}, u_j, u_{j-1})=p_j(r(u_{j+1}, u_j; c)-r(u_j, u_{j-1}; c)),$$

$$r(a, b; c)\equiv r(a+c, b+c)=\frac{a+b}{2}+c-\frac{b-a}{\log((c+b)/(c+a))}, \tag{36}$$

and c is a parameter to be chosen. The purpose of introducing c is two-fold. First, to make the scheme well-defined; second, to lead to a better approximation.

Now the corresponding matrix A in (18) becomes

$$A_n=\begin{pmatrix} 2a+hq_1 & -(a-p_1/2) & & & \\ -(a+p_2/2) & 2a+hq_2 & -(a-p_1/2) & & \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \\ & \cdots\cdots & \cdots\cdots & \cdots\cdots & \\ & & -(a+p_{N-2}/2) & 2a+hq_{N-2} & -(a-p_{N-2}/2) \\ & & & -(a+p_{N-1}/2) & 2a+hq_{N-1} \end{pmatrix}. \tag{37}$$

Denote the determinant of the first j and the last $N-i$ principal determinant by D_j and $D_{i,N-1}$, respectively, and

$$\beta_n \equiv \frac{D_{n-1}}{D_n}, \qquad \beta_{j,N-1} \equiv \frac{D_{j+1,N-1}}{D_{j,N-1}}.$$

Then $D_n = (2a+hq_n)D_{n-1} - \left(a - \frac{p_{n-1}}{2}\right)\left(a + \frac{p_n}{2}\right)D_{n-2}, \quad D_0 \equiv 1, \quad D_1 = 2a+hq_1,$

$$\beta_n = \frac{1}{2a+hq_n - (a-p_{n-1}/2)(a+p_n/2)\beta_{n-1}},$$

and $D_{j,N-1} = (2a+hq_j)D_{j+1,N-1} - \left(a - \frac{p_j}{2}\right)\left(a + \frac{p_{j+1}}{2}\right)D_{j+2,N-1},$

$$D_{N,N-1} \equiv 1, \quad D_{N-1,N-1} = 2a+hq_{N-1}. \tag{38}$$

$$\beta_{j,N-1} = \frac{1}{2a+hq_j - (a-p_j/2)(a+p_{j+1}/2)\beta_{j+1,N-1}}.$$

Lemma 10. *Assume that*

(i) $$a = \frac{\epsilon}{h} \geqslant \frac{1}{2}\|p\|_\infty, \tag{39}$$

(ii) $$q_j \geqslant \frac{1}{2h}(p_{j+1} - p_{j-1}), \quad \textit{for all } j. \tag{40}$$

Then

$$\beta_n \leqslant \frac{1}{a+p_{n+1}/2}, \qquad \beta_{n,N-1} \leqslant \frac{1}{a+p_n/2}, \quad \textit{for all } n < N-1. \tag{41}$$

Remark. As a discrete form of an inequality $q(x) \geqslant p'(x)$, (40) is a sufficient condition of an elliptic form for equation (33).

Proof. The first inequality of (41) is trivial for $n=1$. By induction, suppose it holds for $n-1$. Then from (38) with the condition (40),

$$\beta_n \leqslant \left\{2a + \frac{1}{2}(p_{n+1} - p_{n-1}) - (a - p_{n-1}/2)\right\}^{-1} = \{a + p_{n+1}/2\}^{-1}.$$

The second part of (41) can be proved in the same manner.

Theorem 11. *The inverse matrix of A defined in* (37) *is positive and satisfies* (20) *if* (39) *and* (40) *both hold.*

Proof. In fact, it is not difficult to identify the elements of the inverse matrix

$$\alpha_{i,j}^{-1} = \begin{cases} \prod_{k=j}^{i-1}\left(a + \frac{p_{k+1}}{2}\right)D_{i+1,N-1}\frac{D_{j-1}}{D_{N-1}}, & \text{if } i > j, \\ D_{i+1,N-1}\frac{D_{j-1}}{D_{N-1}}, & \text{if } i = j, \\ \prod_{k=i}^{j-1}\left(a - \frac{p_k}{2}\right)D_{j+1,N-1}\frac{D_{i-1}}{D_{N-1}}, & \text{if } i < j. \end{cases} \tag{42}$$

Hence, in the case $i \geqslant j$,

$$\frac{\alpha_{i,j}^{-1}}{\alpha_{i,j-1}^{-1}} = \frac{D_{j-1}}{(a+p_j/2)D_{j-2}} = \frac{1}{(a+p_j/2)\beta_{j-1}} \geqslant 1;$$

otherwise, $$\frac{\alpha_{i,j-1}^{-1}}{\alpha_{i,j}^{-1}} = \frac{D_{j,N-1}}{(a-p_j/2)D_{j+1,N-1}} > \frac{1}{(a+p_j/2)\beta_{j,N-1}} \geqslant 1.$$

Though the nonlinear term $Q(U^h)$ in (18) whose components are defined in (36), is not homogeneous for U^h, it is homogeneous for $V = U^h + \bar{c}$, where $\bar{c}$ denotes a constant vector. Hence,

$$\{J(Q(V))V\}_j = \{Q(V)\}_j, \quad \text{if } j < N-1, \tag{43}$$

and

$$\sigma_i \equiv \{A^{-1}J(Q(V))V\}_i = p_{N-1}\alpha_{i,N-1}^{-1}R_{N,N-1} - \sum_{j=n}^{N-1}\{p_j\alpha_{i,j}^{-1} - p_{j-1}\alpha_{i,j-1}^{-1}\}r_{j-1/2} - p_n\alpha_{i,n}^{-1}r_{n-1/2},$$

where $$r_{j-1/2} \equiv r(V_j, V_{j-1}), \qquad R_{N,N-1} \equiv r_{N-1/2} + \frac{\partial r_{N-1/2}}{\partial u_N}.$$

It is easily seen that if condition (26) now is changed to (39), Theorem 6 still remains true when $p' \geqslant 0$. Therefore, we obtain an extension of Theorem 6 below.

Theorem 12. *Suppose*

$$p'(x) \geqslant 0, \quad \text{and} \quad q_j \geqslant \frac{1}{2h}(p_{j+1} - p_{j-1}), \quad \text{for all } j.$$

Then the map $P(V) \equiv A^{-1}Q(V)$ is contractive if (39) *holds.*

When $p'(x)$ is negative somewhere, the derivation of conditions for the map $P(V) \equiv A^{-1}Q(V)$ to be contractive is little more complicated. In this case, set

$$\Sigma \equiv \Sigma^+ + \Sigma^-,$$

where the two terms on the right hand side denote a positive part and a non-positive part respectively. Since $0 \leqslant r_j \leqslant r_M$, from Theorem 11,

$$\sum_{j=n}^{i}\{p_j\alpha_{i,j}^{-1} + p_{j-1}\alpha_{i,j-1}^{-1}\}r_{j-1/2} + p_n\alpha_{i,n}^{-1}r_{n-1/2}$$
$$\leqslant r_M(p_i\alpha_{i,i}^{-1} - p_n\alpha_{i,n}^{-1}) + p_n\alpha_{i,n}^{-1}r_{n-1/2} + \sum_{j=n}^{i}{}^{+}\{p_{j-1}\alpha_{i,j-1}^{-1} - p_j\alpha_{i,j}^{-1}\}\{r_M - r_{j-1/2}\}$$
$$< r_M\alpha_{i,i}^{-1}\left\{p_i + \sum_{j=n+1}^{i}{}^{+}(p_{j-1} - p_j)\right\}.$$

Hence $$-\sigma_i < \frac{3}{4} + \frac{1}{2(a + p_{i+1}/2)}\sum_{j=n+1}^{i}{}^{+}(p_{j-1} - p_j).$$

In the same manner as in the derivation in the last section, we obtain the following result.

Theorem 13. *The conclusion in Theorem 12 still holds even if the condition $p' \geqslant 0$ is removed, provided that inequality* (39) *is changed to*

$$a \geqslant \max\left\{\frac{\|p\|_\infty}{2},\ 2\sum_{j=n}^{N-1}{}^{+}(p_{j-1} - p_j) - \frac{1}{2}p_N\right\}. \tag{44}$$

In order to get an error estimate for $u - U^h$, from (28), it is only needed to find a bound for $\|A^{-1}\,\mathrm{Tr}(u)\|_\infty$ when the map is contractive. This reduces to estimating the bound of the truncation error. Substituting the exact solution u in the scheme (35) yields

$$L_h u_j = hf_j + \frac{h^3}{12}\{-\epsilon u^{(4)} + 2pu^{(3)}\}\Big|_\xi, \quad \text{for} \quad j \in I_r,$$

and

$$L_h u_j = hf_j - g_j(u_{j-1}, u_j, u_{j+1}) + \frac{h^3}{12}\left\{-\epsilon u^{(4)} + 2pu^{(3)} - p\left(\frac{u'^2}{c+u}\right)'\right\}\Bigg|_\eta, \text{ for } j \in I_s, \tag{45}$$

where $0 < \xi$, $\eta < 1$.

It has been noted[5] that the exact solution of (33) has a factorization which consists of two parts: one is regular, the other is singular:

$$u(x) = \gamma\{Z(x) + e^{-p(1)(1-x)/\epsilon}\}, \tag{46}$$

where γ is a constant which is bounded uniformly for all $0 < \epsilon < 1$, and

$$|Z(x)|\leqslant C, \quad |Z'(x)|\leqslant C, \quad |Z''(x)|\leqslant C\left\{1+\frac{1}{\epsilon}e^{-\beta(1-x)/\epsilon}\right\},$$

where C is a constant independent of ϵ, and $0<\beta\leqslant p^*$.

Set c in (36) equal to γ which is computable. Because the singularity of the exact solution $u(x)$ is only near $x=1$, the width of the boundary layer in which $u(x)$ has large derivatives is less than k times ϵ. For such a constant k, no matter how small ϵ is, this implies that one can choose $I_s=(1-k\epsilon, 1)$, and on $[0, 1-k\epsilon]$, $u(x)$ and its first and fourth derivatives are uniformly bounded. Hence we only need to consider the maximum error within the boundary layer. From (46), when $h\leqslant\epsilon\|p\|_\infty/2$, in the interval I_s

$$-\epsilon u^{(4)}+p(x)u^{(3)}=c\left\{\left[\frac{p(1)}{\epsilon}\right]^3[p(x)-p(1)]e^{-p(1)(1-x)/\epsilon}-\epsilon Z^{(4)}+p(x)Z^{(3)}\right\}=O(h^{-2}),$$

$$u''-\frac{u'^2}{c+u}=\frac{c^2}{u+c}\left\{\left[\left(\frac{p(1)}{\epsilon}\right)^2e^{-p(1)(1-x)/\epsilon}+Z''\right][e^{-p(1)(1-x)/\epsilon}+Z+c]\right.$$
$$\left.-\left[\frac{p(1)}{\epsilon}e^{-p(1)(1-x)/\epsilon}+Z\right]^2\right\}$$
$$=\frac{c^2}{u+c}e^{-p(1)(1-x)/\epsilon}\frac{p(1)}{\epsilon}\left[\frac{p(1)}{\epsilon}(Z+1)-2\right]+\cdots,$$

$$\left\{u''-\frac{u'^2}{c+u}\right\}'=\frac{c^2}{u+c}e^{-p(1)(1-x)/\epsilon}\left(\frac{p(1)}{\epsilon}\right)^2\left[\frac{p(1)}{\epsilon}(Z+1)-2\right]+\cdots.$$

Since $u(1)=0$, $Z(1)=-1$,

$$u''-\frac{u'^2}{c+u}=O(h^{-1}), \qquad \left\{u''-\frac{u'^2}{c+u}\right\}'=O(h^{-2}).$$

Substituting into (45), we obtain an estimation of the truncation error in the maximum sense:

$$\max_{x\in I_s}|\mathrm{Tr}(x)|=O(h).$$

Note that the corresponding truncation error is $O(1)$ for the case of the usual linear central difference scheme with the same mesh constraint. It means that introducing the semi-linear scheme may give one more order of precision.

Considering that the width of the boundary layer I_s is only $k\epsilon$, the number of knots in using the semi-linear scheme is always less than a constant if the ratio $\frac{h}{\epsilon}$ remains a constant. Hence

$$\|A^{-1}\,\mathrm{Tr}(x)\|_\infty=O(h).$$

Since the error system

$$A(u-U^h)=Q(u)-Q(U)+\mathrm{Tr}(u),$$

so that

$$\|u-U^h\|\leqslant\|A^{-1}(Q(u)-Q(U^h))\|+\|A^{-1}\,\mathrm{Tr}(x)\|,$$
$$\|u-U^h\|<\{1-\|A^{-1}J(Q(u))\|\}^{-1}\|A^{-1}\,\mathrm{Tr}(x)\|,$$

and if $\|A^{-1}J(Q(u))\|\,\|u-U^h\|<1$.

Therefore, we have extended the error estimate Theorem 8 for the general problem (33).

Theorem 14. *The solution of the semi-linear scheme* (35) *converges to the exact solution of the singular perturbation problem* (33) *as h tends to zero if* (40) *holds.*

Moreover, if the ratio of h to ϵ satisfies inequality (39) *or* (44), *according to the conditions of Theorems* 12 *and* 13 *respectively, there exists an error bound such that*

$$\|U^h-u\|_\infty<Ch, \tag{47}$$

where the coefficient C is uniformly bounded for all $\epsilon>0$.

Using the same block matrix technique described in the last section, the mesh constraint (39) or (44) can be limited only in the interval I_s. However, in the regular interval, a larger mesh step is allowed.

5. Computational Results

In this section, three numerical examples are presented to show the effectiveness of the semi-linear scheme. Only uniform mesh sizes are used throughout this section. The Fortran program was run in double-precision, on a DEC-System 2060 computer at Yale. The emphasis is on the comparison among the semi-linear scheme, the corresponding linear central difference scheme, and an upwind method described by Christie *et al.* in[3].

Example 1. Model problem (8)

$$Lu=-\epsilon u''+u'=0, \quad \text{in } (0,1), \quad u(0)=0,\ u(1)=1.$$

Numerical results for the model problem (8) are presented in Tables 1—3, and a comparison of three different methods is given for $\epsilon=1/60$ and $h=1/40$ in the sense of the point-wise error. For the linear central difference scheme, there is a relative error of 36% at the last interior mesh point $x=0.975$. The Upwind Symmetric Quadratics Method has a relative error 1.2% at the same point. The advantage of the semi-linear scheme is clear. Three or seven digits can be improved for an iteration error of $0.1D-3$ or $0.1D-8$, respectively. For a given accuracy, the required CPU time reduced to half in using the semi-linear scheme instead of the usual linear scheme.

Table 1. ($\epsilon=1/60$, $h=1/40$)

x	Theoretical solution	Linear scheme	Semi-linear		Upwind method
			$ei=0.1-3$	$ei=0.1-8$	
0.800	0.000006144	0.000000173	0.000006141	0.000006144	—
0.850	0.000123410	0.000008500	0.000123447	0.000123410	—
0.875	0.000553084	0.000059499	0.000553305	0.000553085	—
0.900	0.002478752	0.000416493	0.002477749	0.002478752	0.0026
0.925	0.011108997	0.002915452	0.011110675	0.011108997	0.0115
0.950	0.049787068	0.020408163	0.049785741	0.049787068	0.0510
0.975	0.223130160	0.142857143	0.223130576	0.223130160	0.2258
1.000	1.0	1.0	1.0	1.0	1.0
Max error of u		$-0.8027-01$	$0.1679-5$	$0.2044-9$	$0.27-2$
Location of max error		0.975	0.925	0.875	0.975
Max error of u'		$0.56+1$	$0.95-4$	$0.11-7$	—
Number of iterations		1	12	22	1
CPU time (seconds)		0.07	0.15	0.26	—

Remark. *ei*—the admissible error for iterations of the semi-linear scheme.

For the same ϵ, Table 2 describes the convergence rate. For an admissible error $0.5D-4$, the semi-linear scheme requires a mesh size h approximately 1/30, while the linear scheme requires a mesh size 1/2000. The total CPU time between these two schemes differs by a factor of 2 to 3. For the first derivative, the difference between the two schemes is even greater.

Table 2. ($\epsilon=1/60$, $ei(SL1)=0.1-4$, $ei(SL2)=0.1-8$)

	M $1/h$	20	30	40	60	640	1920
Max error of u	L	$-0.25+0$	$-0.14+0$	$-0.80-1$	$-0.35-1$	$-0.27-3$	$-0.30-4$
	$SL1$	$-0.46+1$	$0.54-5$	$0.17-5$	$-0.35-6$	$-0.37-9$	$-0.51-12$
	$SL2$	$-0.46+1$	$0.32-9$	$0.20-9$	$-0.51-10$	$0.46-12$	$0.46-16$
Max error of u'	L	$0.70+1$	$0.66+1$	$0.56+1$	$0.39+1$	$0.80-1$	$-0.25-1$
	$SL1$	$-0.10+4$	$0.31-3$	$0.95-4$	$-0.21-4$	$-0.13-6$	$0.50-9$
	$SL2$	$-0.10+4$	$0.19-7$	$0.11-7$	$-0.28-8$	$0.16-9$	$0.51-13$
CPU time	L	0.01	0.02	0.07	0.08	0.46	1.30
	$SL1$	0.93 (100)	0.51 (35)	0.15 (12)	0.16 (7)	0.90 (3)	2.78 (3)
	$SL2$	0.93 (100)	0.74 (71)	0.28 (22)	0.24 (12)	1.10 (4)	3.21 (4)

Remark. 1. The number in the bracket is the number of iterations required to reduce the error to less than the admissible range *ei*.

2. Max $u'=60$.

The influence of the convergence of the semi-linear discretization is described in Table 3, depending upon the ratio of ϵ to h. We consider three cases: $h/\epsilon=2$, 1.5 and 1. According to Theorem 3.2, the iteration converges if $h/\epsilon\leqslant2$. These numerical results satisfy the theory. The iteration is also convergent for $h=2\epsilon$; however, this is not recommended because the rate of convergence is too slow. For a practical choice of $h/\epsilon=1.5$, a higher precision can be obtained with less CPU time.

Hence, to get the same accuracy, one may solve a small semi-linear system instead of the original large linear system. Besides, the errors in the two schemes behave differently. For the linear scheme, when the ratio ϵ/h is constant, the error is also constant independent of the size of ϵ. For the semi-linear scheme the error converges to zero as ϵ does.

Table 3-1. ($h/\epsilon=2$)

	$1/\epsilon$ ei	10 $0.1-4$	20 $0.1-4$	60 $0.1-8$	100 $0.1-12$
Error of u	L	$-0.14+0$	$-0.14+0$	$-0.14+0$	$-0.14+0$
	SL	$-0.45+0$	$-0.96-4$	$0.32-9$	$0.20-12$
Error of u'	L	$0.11+1$	$0.22+1$	$0.66+1$	$0.11+2$
	SL	$0.45+1$	$-0.20-2$	$0.19-7$	$0.18-10$
CPU time	L	0.01	0.01	0.03	0.11
	SL	0.21 (100)	0.35 (100)	0.70 (71)	1.76 (100)

Remark. The number in the bracket is the number of iterations required to reduce the error to less than the admissible range *ei*.

Table 3-2. $(h/\epsilon=3/2)$

	$1/\epsilon$	15	30	60	90
	ei	0.1−4	0.1−4	0.1−8	0.1−12
Error of u	L	−0.80−1	−0.80−1	−0.80−1	−0.80−1
	SL	−0.23−3	0.15−5	0.20−9	0.28−13
Error of u'	L	0.14+1	0.28+1	0.56+1	0.84+1
	SL	−0.39−2	0.43−4	0.11−7	0.24−11
CPU time	L	0.01	0.01	0.03	0.11
	SL	0.03	0.07	0.25	0.70
		(12)	(12)	(22)	(30)

Table 3-3. $(h/\epsilon=1)$

	$1/\epsilon$	5	10	20	60	100
	ei	0.1−4	0.1−4	0.1−4	0.1−8	0.1−12
Error of u	L	−0.35−1	−0.35−1	−0.35−1	−0.35−1	−0.35−1
	SL	−0.49−2	−0.92−3	−0.64−5	−0.51−10	−0.69−14
Error of u'	L	0.32+0	0.64+0	0.13+1	0.38+1	0.64+1
	SL	0.33−1	−0.11−1	−0.15−3	−0.28−8	−0.59−12
CPU time	L	0.01	0.01	0.02	0.06	0.21
	SL	0.01	0.02	0.05	0.23	0.71
		(7)	(7)	(7)	(12)	(17)

Example 2. A linear singular perturbation problem with constant coefficients

$$Lu=-\epsilon u''+u'+(1+\epsilon)u=f(x),\ \text{in}\ (0,\ 1),\ u(0)=u(1)=0,$$

where $f(x)=(1+\epsilon)(a-b)x-\epsilon a-b$, $a=1+e^{-(1+\epsilon)/\epsilon}$, $b=1+e^{-1}$, with exact solution

$$u(x)=e^{-(1+\epsilon)(1-x)/\epsilon}+e^{-x}-a+(a-b)x.$$

It is not difficult to find that the coefficient γ in (46) here is equal to 1 without knowing the exact solution in advance. The result listed in Table 4 shows that iterations converge monotonically if the ratio $h/\epsilon\leqslant 2$ and that the calculated results agree well with the theoretical analysis. More small ϵ is contained, more advantages the semi-linear scheme has. Therefore, with the same accuracy, by using the semi-linear scheme, a large linear system arising from the linear scheme is replaced by a smaller semi-linear system. For instance, when $\epsilon=0.01$, the maximum error in using the semi-linear scheme with $N=60$ (after 13 iterations) is less 50% than the usual linear scheme with $N=200$; meanwhile the ratio of CPU time is 0.27:1.12 (sec.). When $\epsilon=0.001$, the maximum error in using the semi-linear scheme with $N=600$ is less than ten times of the usual linear scheme with $N=2000$, and the ratio of CPU time is 3.22:7.87 (sec.).

Example 3. A semi-linear singular perturbation problem

$$Lu=-\epsilon u''+u'+(1+\epsilon)u=f(x,\ u),\quad \text{in}\ (0,\ 1),\ u(0)=u(1)=0, \tag{48}$$

with the same solution as the problem in Example 2, where

$$f(x,\ u)=a-b-(1+\epsilon)\left\{e^{-x}-u+\frac{c}{u+a-(a-b)x-e^{-x}}\right\},$$

$$a=1+e^{-(1+\epsilon)/\epsilon},\quad b=1+e^{-1},\quad c=e^{2(1+\epsilon)(1-x)/\epsilon}.$$

Table 4-1. (ϵ=0.1, *ei*=1.0−5)

h=1/N		x	max(Er(u))	max(Er(u'))	CPU	*NE*
5	*L*	0.800	0.1935*D*+00	0.1247*D*+01	0.01	1
	SL	0.800	0.1935*D*+00	0.1247*D*+01	0.01	2
6	*L*	0.833	0.1474*D*+00	0.1163*D*+01	0.01	1
	SL	0.833	0.4765*D*−01	0.8202*D*−01	0.03	10
10	*L*	0.900	0.6548*D*−01	0.7837*D*+00	0.01	1
	SL	0.800	0.2255*D*−01	−0.2642*D*+00	0.08	7
20	*L*	0.950	0.1974*D*−01	0.3247*D*+00	0.02	1
	SL	0.850	0.6793*D*−02	−0.1112*D*+00	0.04	4
40	*L*	0.975	0.5493*D*−02	0.1057*D*+00	0.05	1
	SL	0.850	0.1499*D*−02	−0.3305*D*−01	0.10	4

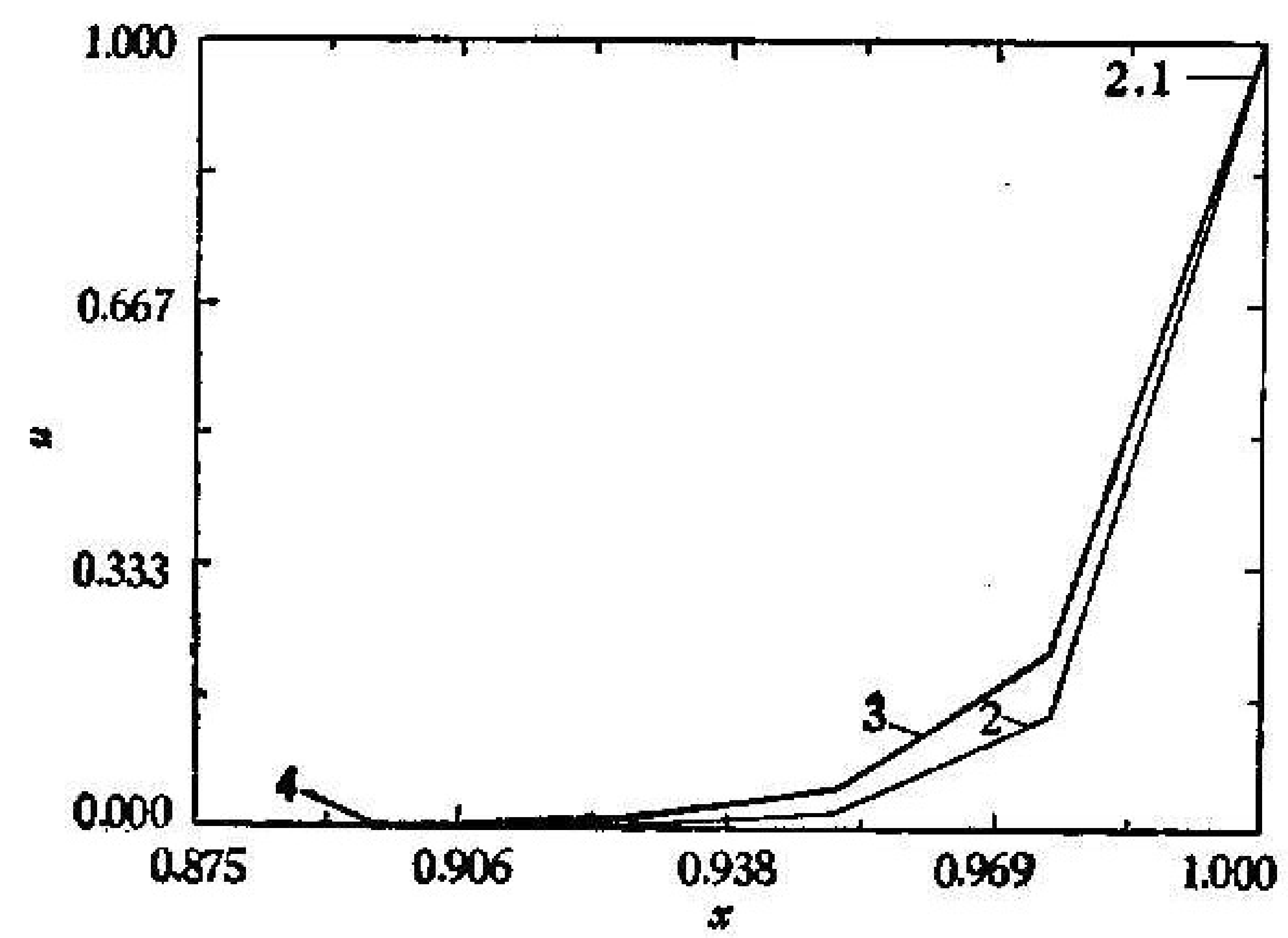
Remark. *NE* is the number of iterations required to reduce the error to less than the admissible range *ei*.

Table 4-2. (ϵ=0.01, *ei*=1.0−5)

h=1/N		x	max(Er(u))	max(Er(u'))	CPU	*NE*
50	*L*	0.980	0.1595*D*+00	0.1116*D*+02	0.15	1
	SL	0.960	0.7958*D*−02	−0.1829*D*+01	0.40	24
60	*L*	0.983	0.1230*D*+00	0.1020*D*+02	0.18	1
	SL	0.967	0.6016*D*−02	−0.1543*D*+01	0.27	13
100	*L*	0.990	0.5561*D*−01	0.6581*D*+01	0.29	1
	SL	0.970	0.2872*D*−02	−0.6184*D*+00	0.43	7
200	*L*	0.995	0.1638*D*−01	0.2624*D*+01	0.57	1
	SL	0.970	0.6339*D*−03	−0.1742*D*+00	0.69	4
400	*L*	0.998	0.4705*D*−02	0.8364*D*+00	1.12	1
	SL	0.970	0.1681*D*−03	−0.5743*D*−01	1.36	4

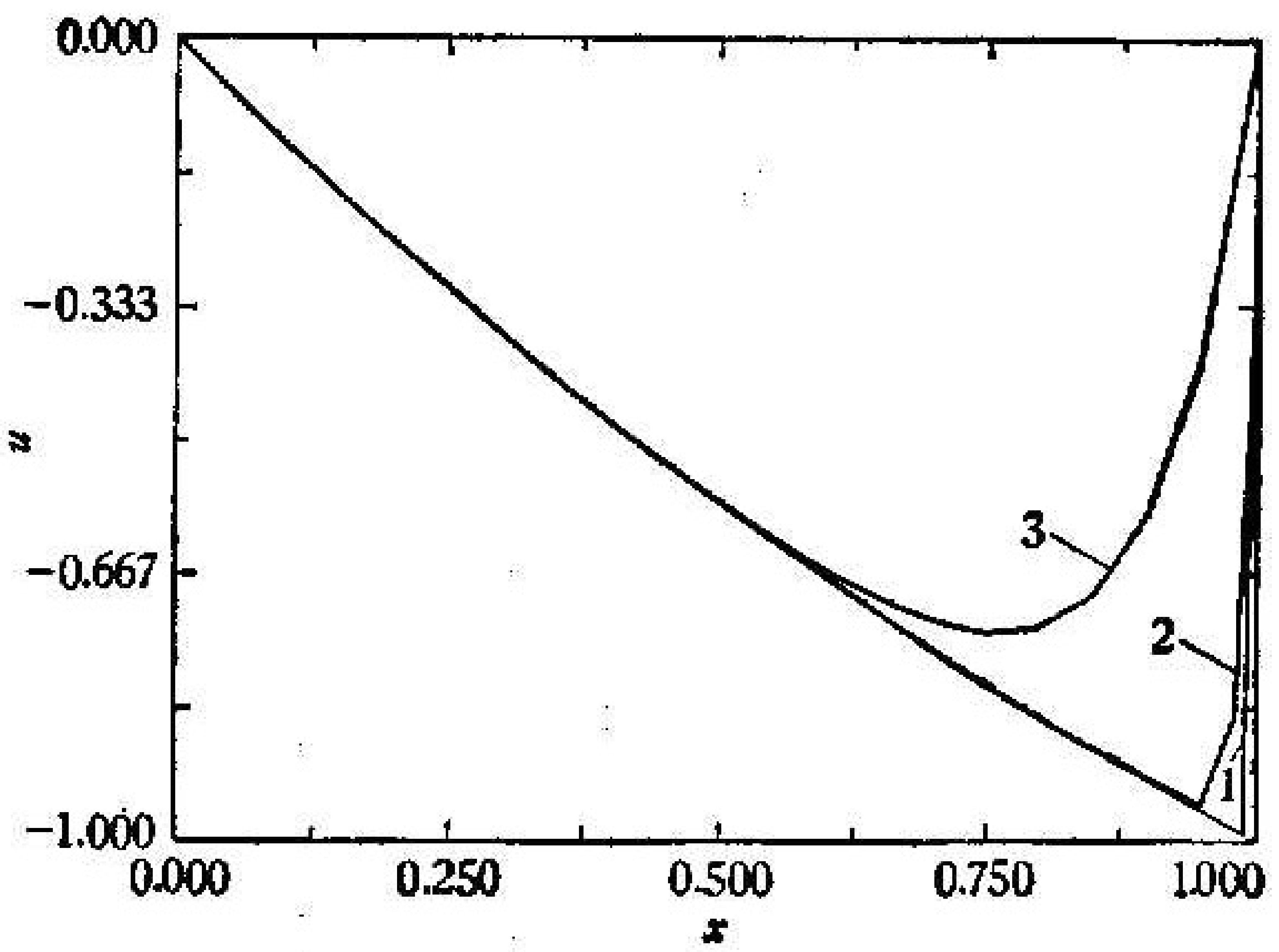
Table 4-3. (ϵ=0.001, *ei*=1.0−5)

h=1/N		x	max(Er(u))	max(Er(u'))	CPU	*NE*
500	*L*	0.998	0.1568*D*+00	0.1102*D*+03	2.05	1
	SL	0.992	−0.1981*D*−02	−0.2046*D*+01	3.42	24
600	*L*	0.998	0.1210*D*+00	0.1006*D*+03	2.47	1
	SL	0.995	0.9313*D*−03	−0.2414*D*+01	3.22	14
1000	*L*	0.999	0.5475*D*−01	0.6459*D*+02	3.93	1
	SL	0.995	0.2977*D*−03	−0.8986*D*+00	5.16	7
1200	*L*	0.999	0.4045*D*−01	0.5219*D*+02	4.77	1
	SL	0.995	0.2150*D*−03	−0.6389*D*+00	5.58	6
2000	*L*	0.999	0.1663*D*−01	0.2565*D*+02	7..87	1
	SL	0.996	0.9066*D*−04	−0.2710*D*+00	9.00	4



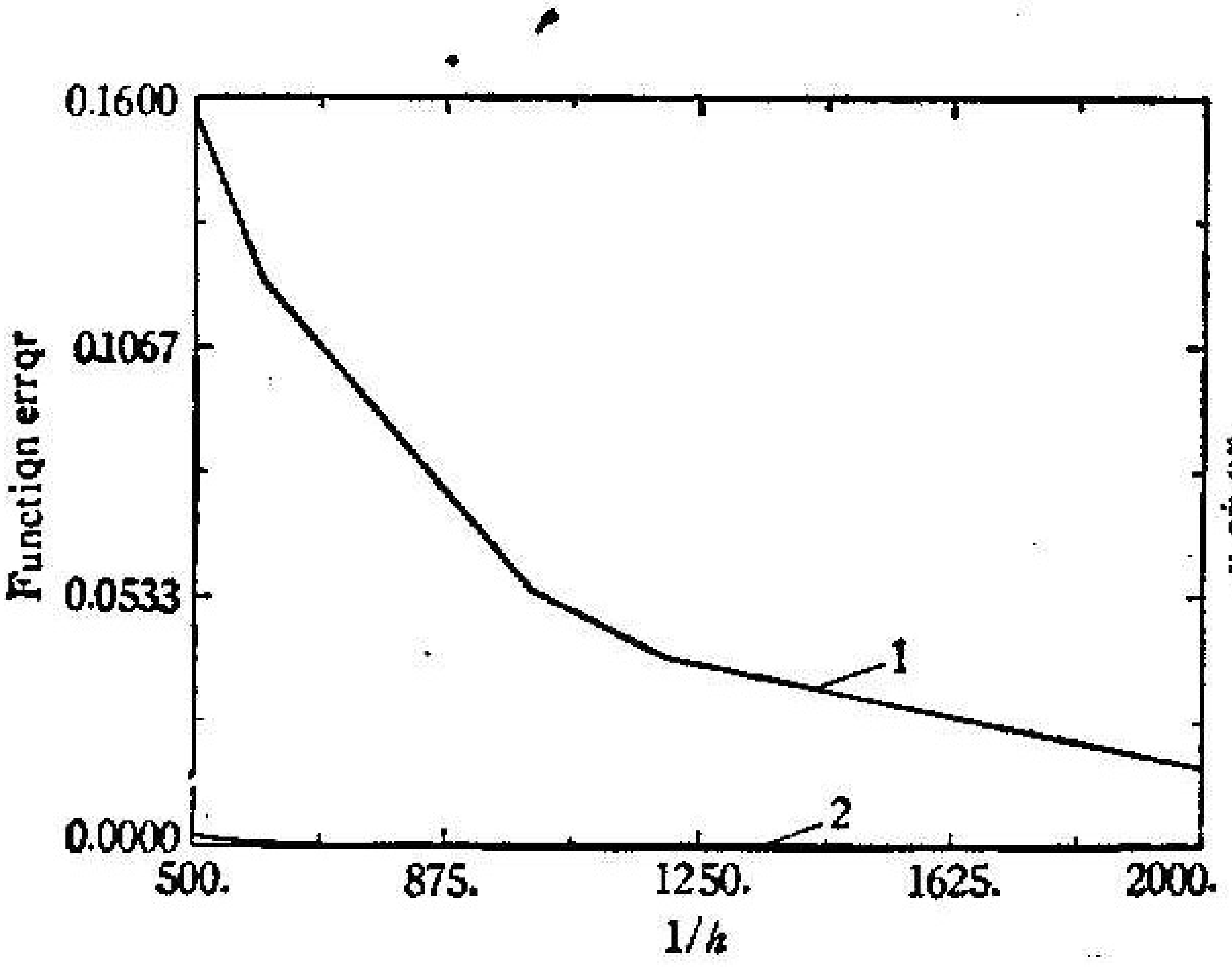
1. Solution $x=0.975$, $u=0.2231$.
2. Linear scheme $u=0.1429$.
3. Semi-linear scheme $u=0.2231$.
4. Upwind $u=0.2258$.

Fig. 1. Three methods comparing ($eps=1/60$, $h=1/40$)



1. $eps=0.1$.
2. $eps=0.01$.
3. $eps=0.001$.

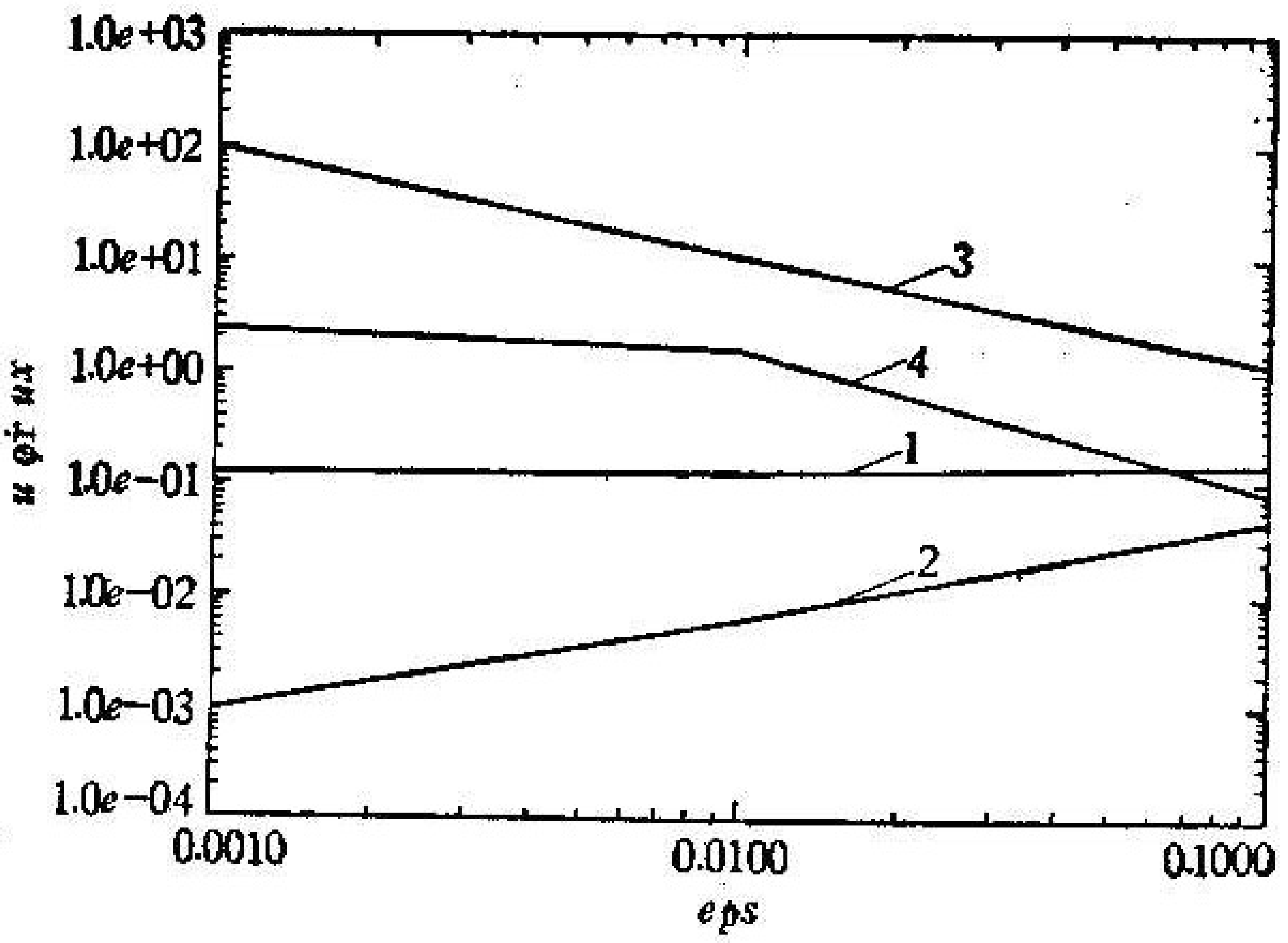
Fig. 2. True solution of examples 2 and 3



1. Linear scheme function error.
2. Semi-linear scheme function error.

Fig. 3. Table 4-3

$eps=0.001$, $1/h=500-2000$



1. Linear scheme function error.
2. Semi-linear scheme function error.
3. Linear scheme first derivative error.
4. Semi-linear scheme first derivative error.

Fig. 4. From Table 4

$h/eps=5/3$, $eps=0.1$, 0.01, 0.001

This example shows more advantages of using *SL*-scheme than *L*-scheme as described in Example 2, because both schemes have to be solved by iteration for this semi-linear problem, and in this case *SL*-scheme requires less CPU time than *L*-scheme even for the same mesh size.

Two different results of two semi-linear schemes are listed in Table 5. *SL*1 scheme is described in section 4. *SL*2 scheme comes from a semi-linear Galerkin method[6]. At any rate, the advantage of both semi-linear schemes over the linear scheme is clear. The stronger the singularity of the solution of the differential equation, the more the advantages.

Table 5-1. $(\epsilon=0.1,\ ei=1.0-5)$

$h=1/N$		x	max(Er(u))	max(Er(u'))	CPU	NE
6	L	0.833	0.1413D+00	0.1163D+01	0.25	100
	$SL2$	0.667	0.1913D−01	0.1468D+00	0.16	11
10	L	0.900	0.1111D+00	0.7837D+00	0.43	100
	SL	0.900	0.2463D−01	−0.2786D+00	0.60	100
	$SL2$	0.800	0.1715D−01	0.2356D+00	0.17	13
20	L	0.950	0.2517D−01	0.3247D+00	0.86	100
	SL	0.900	0.1110D−01	−0.1286D+00	1.12	100
	$SL2$	0.850	0.5012D−02	−0.9715D−01	0.32	13
40	L	0.975	0.6984D−02	0.1057D+00	0.30	14
	SL	0.900	0.2445D−02	−0.4427D−01	0.35	14
	$SL2$	0.825	0.1004D−02	−0.2500D−01	0.70	13
80	L	0.987	0.1812D−02	0.3023D−01	0.63	14
	SL	0.900	0.6265D−03	−0.1872D−01	0.73	14
	$SL2$	0.837	0.2561D−03	−0.7389D−02	1.33	13

Table 5-2. $(\epsilon=0.01,\ ei=1.0-5)$

$h=1/N$		x	max(Er(u))	max(Er(u'))	CPU	NE
50	L	0.980	0.1391D+00	0.1116D+02	4.33	100
	$SL1$	0.960	0.8372D−02	−0.1886D+01	4.72	100
	$SL2$	0.960	0.7927D−02	−0.1662D+01	11.67	100
60	L	0.983	0.1255D+00	0.1020D+02	7.37	100
	$SL2$	0.967	0.5378D−02	−0.1375D+01	1.47	10
100	L	0.990	0.5766D−01	0.6581D+01	1.08	10
	$SL1$	0.970	0.3158D−02	−0.6582D+00	1.11	9
	$SL2$	0.960	0.2617D−02	−0.5572D+00	2.20	9
200	L	0.995	0.1733D−01	0.2624D+01	2.05	9
	$SL1$	0.975	0.6970D−03	−0.1894D+00	2.14	9
	$SL2$	0.965	0.5763D−03	−0.1500D+00	4.44	9
400	L	0.998	0.4812D−02	0.8364D+00	4.13	9
	$SL1$	0.973	0.1845D−03	−0.6287D−01	4.51	9
	$SL2$	0.968	0.1536D−03	−0.4842D−01	8.86	9

Table 5-3. $(\epsilon=0.001,\ ei=1.0-5)$

$h=1/N$		x	max (Er(u))	max (Er(u'))	CPU	NE
500	L	0.998	0.2943D−01	0.1102D+03	58.29	100
	$SL1$	0.994	0.1020D−02	−0.3495D+01	59.29	100
	$SL2$	0.994	0.1129D−02	−0.3240D+01	153.63	100
600	L	0.998	0.1211D+00	0.1006D+03	71.83	100
	$SL2$	0.995	0.9154D−03	−0.2334D+01	23.22	11
1000	L	0.999	0.5494D−01	0.6459D+02	14.16	9
	$SL1$	0.995	0.3019D−03	−0.9171D+00	14.05	9
	$SL2$	0.995	0.2907D−03	−0.8533D+00	28.66	9
2000	L	0.999	0.1667D−01	0.2565D+02	27.56	9
	$SL1$	0.996	0.9201D−04	−0.2771D+00	27.70	9
	$SL2$	0.995	0.8845D−04	−0.2594D+00	57.68	9

Acknowledgement. The author would like to thank Professor Martin H. Schultz for his many valuable comments and Dr. Ding Lee for his help in improving this English manuscript.

References

[1] J. W. Barrett, K. W. Morton, Optimal finite element solutions to diffusion-convection problems in one dimension, *Int. J. Num. Meths. in Engng.*, v. 15, 1980, 1457—1474.

[2] J. Christie, D. F. Griffith, A. R. Mitchell, O. C. Zienkienwicz, Finite element methods for second order differential equations with significant first derivatives, *Int. J. Num. Meths. in Engng.*, v. 10, 1976, 1389—1396.

[3] J. Christie, A. R. Mitchell, Upwinding of high order Galerkin methods in conduction-convection problems, *Int. J. Num. Meths. in Engng.*, v. 12, 1978, 1764—1771.

[4] J. C. Heinrich, P. S. Huyakorn, O. C. Zienkienwicz, A. R. Mitchell, An "Upwind" finite element scheme for two-dimensional convective transport equation, *Int. J. Num. Meths. in Engng.*, v. 11, 1977, 131—143.

[5] R. B. Kellogg, Honde Han, *The Finite Element Method for a Singular Perturbation Problem Using Enriched Subspaces*, Technical Note #BN-978, University of Maryland, 1981.

[6] Jiachang Sun, *A Galerkin Method on Nonlinear Subsets and Its Application to a Singular Perturbation Problem*, Department of Computer Science, Yale University, Technical Report, #217, 1982.