

NUMERICAL METHODS FOR SINGULAR PERTURBATION PROBLEMS*

BARBRO KREISS† AND HEINZ-OTTO KREISS†

Dedicated to Robert D. Richtmyer on the occasion of his seventieth birthday

Abstract. Consider the two-point boundary value problem for a stiff system of ordinary differential equations. An adaptive method to solve these problems even when turning points are present is discussed.

1. Introduction. Consider a system of ordinary differential equations,

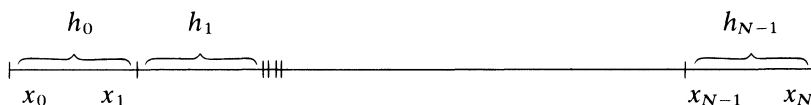
$$(1.1) \quad \frac{dy}{dx} = A(x)y + F(x), \quad 0 \leq x \leq 1,$$

with n linearly independent boundary conditions,

$$(1.2) \quad R_0 y(0) + R_1 y(1) = g.$$

Here $y' = (y^{(1)}, \dots, y^{(n)})^1$ is a vector function with n components, and R_0, R_1 and $A(x) \in C^{1,2}$ are $n \times n$ matrices.

We want to solve the above problem by difference approximations. For that reason we divide the x -axis into subintervals of variable length h_j with grid points $x_0 = 0$, $x_\nu = \sum_{j=0}^{\nu-1} h_j$, $\nu = 1, 2, \dots, N$, $x_N = 1$, and denote by $u_\nu = u(x_\nu)$ vector functions defined on the grid



Let $h = \max_j h_j$. The case where we can choose h so small that $h|A| \ll 1$ has been treated many times before. Our aim is to treat the case $h|A| \gg 1$; i.e., we want to discuss methods for stiff equations. There are essentially two difficulties. (1) The matrix A has large eigenvalues of both signs. (2) There are turning points, i.e., these large eigenvalues are changing signs.

2. A simple example. Consider the system

$$(2.1) \quad \varepsilon \frac{dy}{dx} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & +1 \end{pmatrix} y = Ay, \quad 0 \leq x \leq 1, \quad \varepsilon > 0,$$

with boundary conditions

$$(2.2) \quad \begin{aligned} y^{(1)}(0) + \alpha_0 y^{(2)}(0) + \beta_0 y^{(3)}(0) &= g_1, \\ y^{(2)}(0) + \gamma_0 y^{(3)}(0) &= g_2, \\ y^{(3)}(1) + \alpha_1 y^{(2)}(0) + \beta_1 y^{(1)}(1) &= g_3. \end{aligned}$$

* Received by the editors June 9, 1980.

† California Institute of Technology, Pasadena, California 91125.

¹ If y is a vector then y' denotes its transpose and y^* its adjoint. The vector norm is defined by $|y| = \max |y^{(i)}|$. Similar notations hold for matrices, for example $|A| = \sup_y |Ay|/|y|$. Furthermore, for vector functions $\|y(x)\| = \max_{0 \leq x \leq 1} |y(x)|$ denotes the maximum norm.

² $A(x) \in C^j$ if the elements of A are j times continuously differentiable.

Here $\varepsilon > 0$ is a small positive constant. The general solution of (2.1) is given by

$$\begin{aligned} y^{(1)}(x) &= e^{-x/\varepsilon} y^{(1)}(0), \\ y^{(2)}(x) &\equiv y^{(2)}(0), \\ y^{(3)}(x) &= e^{(x-1)/\varepsilon} y^{(3)}(1). \end{aligned}$$

Introducing it into the boundary conditions gives us

$$\begin{pmatrix} 1 & \alpha_0 & \beta_0 e^{-1/\varepsilon} \\ 0 & 1 & \gamma_0 e^{-1/\varepsilon} \\ \beta_1 e^{-1/\varepsilon} & \alpha_1 & 1 \end{pmatrix} \begin{pmatrix} y^{(1)}(0) \\ y^{(2)}(0) \\ y^{(3)}(1) \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}.$$

Therefore, neglecting terms which are exponentially small, we obtain

$$y^{(2)}(0) \approx g_2, \quad y^{(1)}(0) \approx g_1 - \alpha_0 g_2, \quad y^{(3)}(1) \approx g_3 - \alpha_1 g_2.$$

This shows that away from the boundary layers the solution of (2.1), (2.2) is smooth.

We approximate the above problem by a standard difference approximation, namely the trapezoidal rule

$$(2.3) \quad \frac{u_{\nu+1} - u_\nu}{h} = A \left(\frac{u_{\nu+1} + u_\nu}{2} \right), \quad \nu = 0, 1, 2, \dots, N-1$$

on a uniform mesh; i.e., $h_i = h$. The desired solution shall satisfy the boundary conditions (2.2). The general solution of (2.3) is given by

$$(2.4) \quad u_\nu^{(1)} = \kappa^\nu u_0^{(1)}, \quad u_\nu^{(2)} = u_0^{(2)}, \quad u_{N-\nu}^{(3)} = \kappa^\nu u_N^{(3)},$$

where

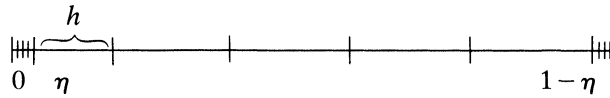
$$\kappa = \frac{2\varepsilon - h}{2\varepsilon + h}.$$

Introducing (2.4) into the boundary conditions (2.2) gives us

$$(2.5) \quad \begin{pmatrix} 1 & \alpha_0 & \beta_0 \kappa^N \\ 0 & 1 & \gamma_0^N \\ \beta_1 \kappa^N & \alpha_1 & 1 \end{pmatrix} \begin{pmatrix} u_0^{(1)} \\ u_0^{(2)} \\ u_N^{(3)} \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}.$$

If h is so small that $h/2\varepsilon \ll 1$, then $\kappa \sim e^{-h/\varepsilon}$ and it is obvious that the solution of (2.3) behaves like the solution of the differential equations. However, if $h/2\varepsilon \gg 1$ then $\kappa \sim -1$, and the solution of (2.5) does not approximate the solution of the differential equations at all, because $u_\nu^{(1)}, u_\nu^{(3)}$ oscillate wildly. In particular, if $g_1 = \alpha_0 g_2, g_3 = \alpha_1 g_2$, then $y^{(1)}(x), y^{(3)}(x)$ are exponentially small and the solution of the differential equation is smooth up to the boundary. The corresponding solution of the difference approximation is still wildly oscillating. There are two ways to overcome this difficulty.

(1) We use the trapezoidal rule for all components but introduce in the boundary layers $0 \leq x \leq \eta, 1 - \eta \leq x \leq 1$, $\eta = O(\varepsilon |\log \varepsilon|)$, new "stretched" independent variables, such that the boundary layer solutions are smooth functions of these new variables. Then we use a uniform grid in the new variables. In the old variables we get a nonuniform mesh which is so fine that the boundary layer solutions change slowly from one point to the next. Away from the boundary layers, we use a uniform mesh, i.e., $h_\nu = h$ for $\eta \leq x \leq 1 - \eta$. This technique was used extensively by C. Pearson [5].



In the boundary layer regions, the differential equations are well approximated by the difference equations. Therefore, for $h \gg \varepsilon$, we get essentially

$$u^{(1)} \approx \begin{cases} e^{-x_\nu/\varepsilon} u_0^{(1)} & \text{for } 0 \leq x_\nu \leq \eta, \\ \pm(-1)^\nu u^{(1)}(\eta) & \text{for } \eta \leq x_\nu \leq 1 - \eta, \quad u_\nu^{(2)} \equiv u_0^{(2)}, \\ e^{-(x_\nu - 1 + \eta)/\varepsilon} u^{(1)}(1 - \eta) & \text{for } 1 - \eta \leq x_\nu \leq 1, \end{cases}$$

and correspondingly for $u_\nu^{(3)}$. Observe that $u_\nu^{(1)}$, $u_\nu^{(3)}$ oscillate wildly in the interval $\eta \leq x \leq 1 - \eta$. However, if η is chosen sufficiently large, the amplitude is so small that it has no effect. It is clear that now the solution of the difference equations approximates the solution of the differential equations well.

The drawback of this method is that we have to use the refinement even if the boundary conditions are such that the actual solution of the differential equation is smooth up to the boundary. The same is true if turning points are present. We have to construct the mesh such that all solutions through the turning point become smooth. For nonlinear problems, one often does not know the position of the turning point and the behavior of the solution. This makes the construction of the mesh rather difficult.

(2) Instead of using the trapezoidal rule for all components, we could use one-sided schemes for the first and last component, for example,

$$(2.6) \quad \varepsilon \frac{u_{\nu+1}^{(1)} - u_\nu^{(1)}}{h} = -u_{\nu+1}^{(1)}, \quad \varepsilon \frac{u_{\nu+1}^{(3)} - u_\nu^{(3)}}{h} = u_\nu^{(3)}, \quad \varepsilon \frac{u_{\nu+1}^{(2)} - u_\nu^{(2)}}{h} = 0.$$

The general solution of (2.6) has the form (2.4) and $u_0^{(j)}$, $j = 1, 2, 3$ are given by (2.5), but now

$$\kappa = \frac{1}{1 + h/\varepsilon}.$$

If now $\varepsilon \ll h$, we still have $|\kappa| \ll 1$ and it is clear that the solution of the difference equations resembles the solution of the differential equations. This technique can be used for rather general systems. However, the boundary layers are not resolved adequately. In general, this will result in an additional error of order $O(\varepsilon)$. To obtain accurate results, one has to resolve the boundary layers (see for example [5]), or a combination of the asymptotic expansion of the boundary layer solution and the approximation (2.6) (see for example [4]).

One can refine the scheme (2.6) considerably, and we shall do this in the next section.

The main problem for the numerical solution of singular perturbation problems is to find the mesh on which the solution varies slowly. There are two possible ways to do this.

(1) One can use the behavior of the coefficients of the differential equation to determine the variation of the solution. This approach has been discussed in [1], [2]. An extended version of [2] is under preparation [3].

(2) In this paper, we want to refine the mesh adaptively. Starting with some mesh, one computes the numerical solution and adds or deletes mesh points according to the variation of the numerical solution. The technique used is a considerable improvement over earlier test calculations done by N. Nichols and H. O. Kreiss during the summer of 1974 in Stanford.

In this paper, we present only the results of a number of calculations. For the theory, we refer to [5].

3. Difference approximation for a scalar equation. Consider a scalar equation

$$(3.1) \quad \frac{dy}{dx} = a(x)y + f(x), \quad 0 \leq x,$$

with initial data

$$y(0) = y_0.$$

Here $a(x), f(x)$ are complex-valued functions, which can be large. We are not interested in the oscillatory stiff case. Therefore, we assume that there are constants ρ, c of "moderate size" and that

$$(3.2) \quad \rho |\operatorname{Re} a| \geq |a(x)| - C.$$

We approximate (3.1) by

$$(3.3) \quad \frac{u_{\nu+1} - u_{\nu}}{h_{\nu}} = \alpha_{\nu} a_{\nu} u_{\nu} + (1 - \alpha_{\nu}) a_{\nu+1} u_{\nu+1} + \alpha_{\nu} f_{\nu} + (1 - \alpha_{\nu}) f_{\nu+1},$$

which can be written as

$$(3.4) \quad u_{\nu+1} = A_{\nu} u_{\nu} + h_{\nu} F_{\nu},$$

where

$$A_{\nu} = \frac{1 + h_{\nu} \alpha_{\nu} a_{\nu}}{1 - h_{\nu} (1 - \alpha_{\nu}) a_{\nu+1}}, \quad F_{\nu} = \frac{\alpha_{\nu} f_{\nu} + (1 - \alpha_{\nu}) f_{\nu+1}}{1 - h_{\nu} (1 - \alpha_{\nu}) a_{\nu+1}}.$$

We want to choose the α_{ν} in such a way that the following conditions are satisfied.

- (1) The method is second-order accurate for $|h_{\nu} a_{\nu}| \ll 1$.
- (2) If $\operatorname{Re} a_{\nu} \ll -1$, then the solutions of the homogeneous equation $v_{\nu+1} = A_{\nu} v_{\nu}$ decay rapidly. This we express by

$$|A_{\nu}| \leq (1 + h_{\nu} \gamma_{\nu})^{-1},$$

where $\gamma_{\nu} = \sigma \max(|\operatorname{Re} a_{\nu}|, |\operatorname{Re} a_{\nu+1}|)$, $\sigma = \text{const.} > 0$.

- (3) If $\operatorname{Re} a_{\nu} \gg 1$, then the solutions of $v_{\nu+1} = A_{\nu} v_{\nu}$ increase rapidly. This we express by

$$|A_{\nu}| \geq 1 + h_{\nu} \gamma_{\nu}.$$

- (4) The α_{ν} are Lipschitz continuous functions of $a_{\nu} h_{\nu}$.

An easy calculation shows that these conditions are satisfied if we choose α_{ν} in the following way:

- I. If $\operatorname{Re} h_{\nu} a_{\nu} \leq 0$ and $\operatorname{Re} h_{\nu} a_{\nu+1} \leq 0$, then

$$\alpha_{\nu} = \begin{cases} \frac{1}{2} & \text{if } |\operatorname{Re} h_{\nu} a_{\nu}| \leq 1, \\ \frac{1}{2|\operatorname{Re} h_{\nu} a_{\nu}|} & \text{if } |\operatorname{Re} h_{\nu} a_{\nu}| > 1. \end{cases}$$

- II. If $\operatorname{Re} h_{\nu} a_{\nu} \geq 0$ and $\operatorname{Re} h_{\nu} a_{\nu+1} \geq 0$, then

$$\alpha_{\nu} = \begin{cases} \frac{1}{2} & \text{if } |\operatorname{Re} h_{\nu} a_{\nu+1}| \leq 1, \\ 1 - \frac{1}{2 \operatorname{Re} h_{\nu} a_{\nu+1}} & \text{if } |\operatorname{Re} h_{\nu} a_{\nu+1}| > 1. \end{cases}$$

III. If $\operatorname{Re} h_\nu a_\nu > 0$ and $\operatorname{Re} h_\nu a_{\nu+1} < 0$, then $\alpha_\nu = \frac{1}{2}$.

IV. If $\operatorname{Re} h_\nu a_\nu < 0$ and $\operatorname{Re} h_\nu a_{\nu+1} > 0$, then introduce a new point x_ν^* with $x_\nu \leq x_\nu^* < x_{\nu+1}$ where $\operatorname{Re} h_\nu a(x_\nu^*) = 0$. Then I or II is applicable.

Remark. For linear equations, the condition that α_ν is a Lipschitz continuous function of $h_\nu a_\nu$ is not necessary. One could use the standard procedure

$$\alpha_\nu = \begin{cases} 1 & \text{if } \operatorname{Re} h_\nu a_\nu > 1, \\ \frac{1}{2} & \text{if } |\operatorname{Re} h_\nu a_\nu| \leq 1, \\ 0 & \text{if } \operatorname{Re} h_\nu a_\nu < -1. \end{cases}$$

This is the procedure we proposed in [4]. However, for nonlinear equations we use Newton's method and the discontinuous change of the formulas can cause convergence problems. Also, if one wants to use Richardson extrapolation one needs an even smoother transition.

We shall now describe the procedure to refine the mesh. Assume we have computed the solution of (3.3) on a mesh $0 = x_1 < x_2 < \cdots < x_N = 1$. Let

$$\mathcal{J}_{\nu,1} = \frac{u_\nu - u_{\nu-1}}{h_\nu}, \quad \mathcal{J}_{\nu,2} = \frac{\mathcal{J}_{\nu+1,1} - \mathcal{J}_{\nu,1}}{h_\nu + h_{\nu+1}},$$

denote the first and second divided differences, respectively. Under reasonable assumptions (see [5]) one can prove that the error $\max_\nu |y(x_\nu) - u_\nu|$ can be estimated by $\max_\nu M[x_{\nu-1}, x_\nu, x_{\nu+1}]$, where

$$M[x_{\nu-1}, x_\nu, x_{\nu+1}] = (h_\nu^2 + h_{\nu+1}^2)(|\mathcal{J}_{\nu,2}| + |\mathcal{J}_{\nu,1}| + |u_\nu|).$$

Therefore the strategy is to add points if $M[x_{\nu-1}, x_\nu, x_{\nu+1}]$ is too large and to delete points if $M[x_{\nu-1}, x_\nu, x_{\nu+1}]$ is very small. In detail, we proceed as follows. Let Δ denote a threshold constant, and assume we have constructed the mesh for $x \leq x_{\nu-1}$.

If $M[x_{\nu-1}, x_\nu, x_{\nu+1}] > \Delta$ then we add the points $x_{\nu-1} + \frac{1}{2}h_{\nu-1}$, $x_\nu + \frac{1}{2}h_\nu$.

If $M[x_{\nu-1}, x_\nu, x_{\nu+1}] < \frac{1}{3}\Delta$, then we investigate $M[x_{\nu-1}, x_{\nu+1}, x_{\nu+2}]$. If also $M[x_{\nu-1}, x_{\nu+1}, x_{\nu+2}] < \frac{1}{3}\Delta$, then we delete the point x_ν and investigate $M[x_{\nu-1}, x_{\nu+2}, x_{\nu+3}]$. If $M[x_{\nu-1}, x_{\nu+2}, x_{\nu+3}] < \frac{1}{3}\Delta$ also, then we delete $x_{\nu+1}$, etc.

This procedure gives us a new mesh. However, numerical experience has shown that one should not change the mesh size too fast. Therefore, we add more points such that

$$(3.5) \quad \frac{1}{3} \leq \left| \frac{h_\nu}{h_{\nu+1}} \right| \leq 3.$$

This is done by the following procedure.

If $|h_\nu/h_{\nu+1}| < \frac{1}{3}$, then we add the point $x_{\nu+1} + \frac{1}{2}h_{\nu+1}$.

If $|h_\nu/h_{\nu+1}| > 3$, then we add the point $x_\nu + \frac{1}{2}h_\nu$. This process is repeated until (3.5) is satisfied everywhere.

The next step is to calculate the solution of (3.3) on the new mesh.

4. Difference approximations for systems. In applications, the systems are often of the form

$$(4.1) \quad \frac{dy}{dx} = \left(\frac{1}{\epsilon} A(x) + B(x) \right) y + F(x).$$

Here $\varepsilon > 0$ is a small constant and

$$A(x) = \begin{pmatrix} a_{11}(x) & a_{12}(x) & \cdots & a_{1n}(x) \\ a_{21}(x) & & \cdots & a_{2n}(x) \\ & & \cdots & \\ a_{n1}(x) & & \cdots & a_{nn}(x) \end{pmatrix},$$

$$B(x) = \begin{pmatrix} b_{11}(x) & b_{12}(x) & \cdots & b_{1n}(x) \\ b_{21}(x) & & \cdots & b_{2n}(x) \\ & & \cdots & \\ b_{n1}(x) & & \cdots & b_{nn}(x) \end{pmatrix}$$

are smooth functions of x .

If $A(x)$ is upper triangular, i.e., $a_{ij} = 0$ for $i > j$, then we can write (4.1) formally as n scalar equations,

$$(4.2) \quad \frac{dy^{(i)}}{dx} = \left(\frac{1}{\varepsilon} a_{ii} + b_{ii} \right) y^{(i)} + G^{(i)},$$

where

$$G^{(i)} = \frac{1}{\varepsilon} \sum_{j=i+1}^n a_{ij} y^{(j)} + \sum_{\substack{j=1 \\ j \neq i}}^n b_{ij} y^{(j)} + F^{(i)}.$$

Thus, we can use the scheme and the refinement procedure of the last section for every equation in (4.2).

If $A(x)$ is not upper triangular, then we have to transform $A(x)$ to upper triangular form. This can be done analytically or by the Q - R method which is economical. Assume that we want to calculate the solution of the difference equation on a mesh $0 = x_1 < x_2 < \cdots < x_N = 1$. Then we construct unitary matrices U_i such that

$$U_i^* A(x_i) U_i = \begin{pmatrix} \tilde{a}_{11}(x_i) & & & \tilde{a}_{1n}(x_i) \\ 0 & \tilde{a}_{22}(x_i) & \cdots & \tilde{a}_{2n}(x_i) \\ & & \cdots & \\ 0 & & & \tilde{a}_{nn}(x_i) \end{pmatrix}.$$

In every interval $x_i \leq x \leq x_{i+1}$, we introduce a new variable by

$$y = U\tilde{y}, \quad U(x) = U_i + (U_{i+1} - U_i) \frac{x - x_i}{x_{i+1} - x_i},$$

and obtain, from (4.1),

$$\frac{d\tilde{y}}{dx} = U^*(x) \left(\frac{1}{\varepsilon} A(x) + B(x) \right) U(x) \tilde{y} - U^* \frac{dU}{dx} \tilde{y} + U^* F.$$

Now $U^* A U$ is upper triangular in the mesh points and we can apply the previous method.

5. Numerical examples. In this section, we consider second-order equations

$$(5.1) \quad \varepsilon \frac{d^2 y}{dx^2} = \frac{d(a(x)y)}{dx} + b(x)y, \quad -c \leq x \leq d, \quad y(-c) = \alpha, \quad y(d) = \beta.$$

We write them as a first-order system by introducing a new variable

$$\frac{dv}{dx} = b(x)y.$$

Then we can integrate (5.1) and obtain

$$(5.2) \quad \frac{dy}{dx} = \left(\frac{a(x)}{\varepsilon} \right) y + f_1, \quad f_1 = \varepsilon^{-1}v, \quad \frac{dv}{dx} = f_2, \quad f_2 = by.$$

We think of the system as two scalar equations and apply the method developed earlier. In particular, the second equation will always be approximated by

$$(5.3) \quad \frac{v_{\nu+1} - v_{\nu}}{f_{\nu}} = \frac{1}{2}(b_{\nu+1}y_{\nu+1} + b_{\nu}y_{\nu}).$$

We present in Figs. 1-10 a number of computer printouts of our results. The notation is as follows (compare with [3]).

$$\text{EPS} * \text{D2Y}/\text{DX}^2 = \text{D}((-X^3 + X/2) * Y)/\text{DX} + (3 * X^2 + 0.2) * Y = 0$$

stands for

$$\varepsilon \frac{d^2 y}{dx^2} = \frac{d((-x^3 + x/2)y)}{dx} + (3x^2 + 0.2)y.$$

EPS = 1.0E-5 means $\varepsilon = 10^{-5}$.

TOL = 0.1 is of no consequence for linear equations. For nonlinear equations, it tells the machine to stop the Newton iteration when the residue is smaller than the tolerance.

DELTA = 0.03 denotes the value of the threshold constant in the mesh refinement procedure.

ITERATIONS is of no consequence for linear equations. For nonlinear problems it counts the number of Newton iterations.

REFINEMENTS denotes the number of mesh refinements.

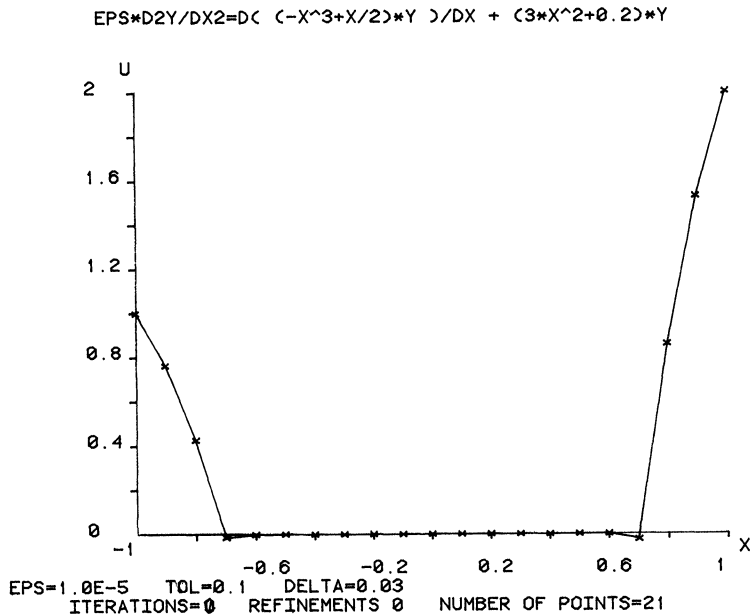
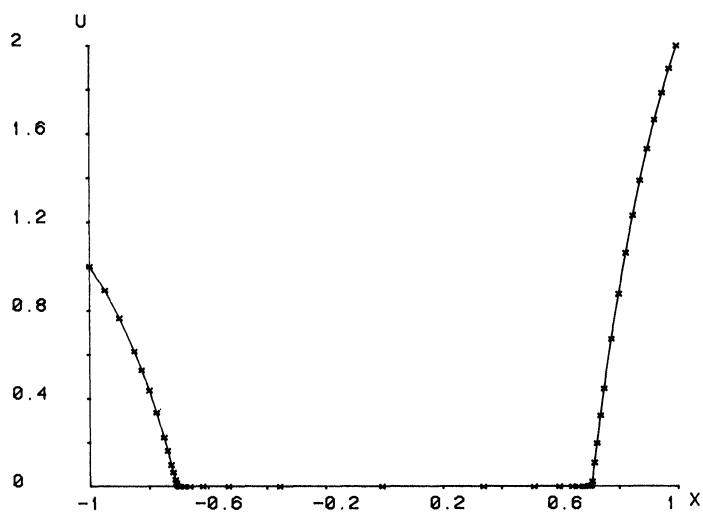


FIG. 1

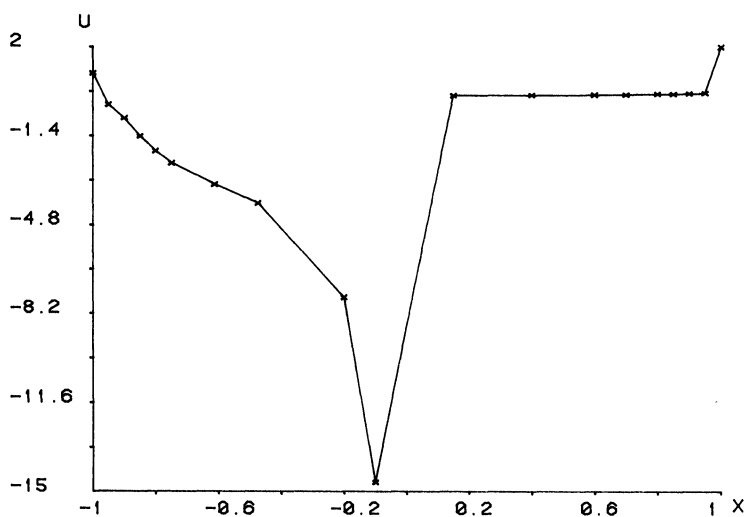
$$\text{EPS} * D^2 Y / DX^2 = D(-X^3 + X/2) * Y / DX + (3 * X^2 + 0.2) * Y$$



EPS=1.0E-5 TOL=0.1 DELTA=0.03
ITERATIONS= 0 REFINEMENTS 9 NUMBER OF POINTS=46

FIG. 2

$$\text{EPS} * D^2 Y / DX^2 = D(-(\sin(\pi * X))^2 * Y) / DX + (-X + 1 + \pi * \sin(2 * \pi * X)) * Y$$



EPS=1.0E-5 TOL=0.1 DELTA=0.08
ITERATIONS=0 REFINEMENTS 1 NUMBER OF POINTS=19

FIG. 3

$$\text{EPS} \cdot D^2Y/DX^2 = D(-(\sin(\pi X))^2 Y)/DX + (-X + 1 + \pi \sin(2\pi X)) Y$$

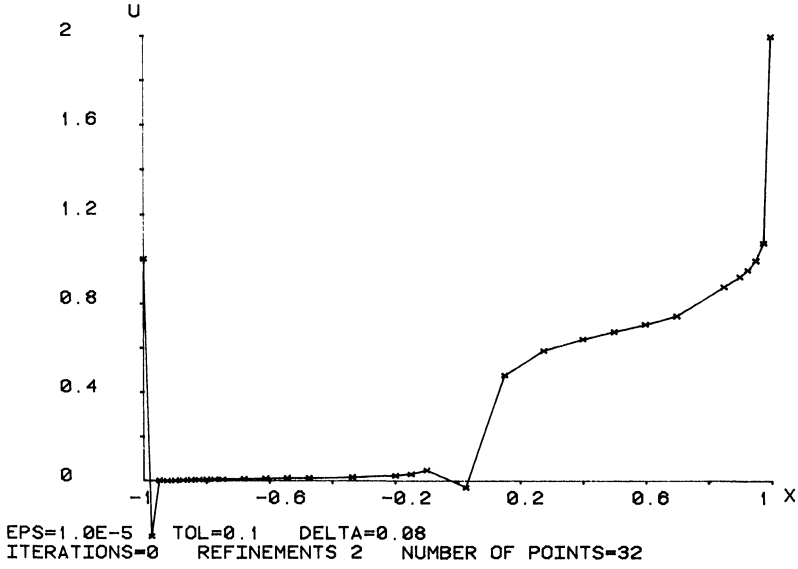


FIG. 4

$$\text{EPS} \cdot D^2Y/DX^2 = D(-(\sin(\pi X))^2 Y)/DX + (-X + 1 + \pi \sin(2\pi X)) Y$$

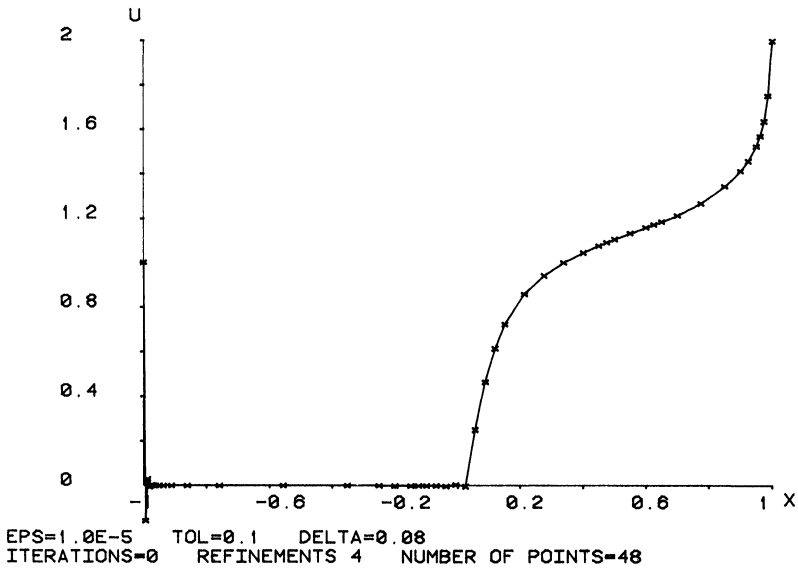


FIG. 5

$$\text{EPS} \cdot D^2Y/DX^2 = D(-(\sin(\pi X))^2 Y)/DX + (-X+1+\pi \sin(2\pi X))Y$$

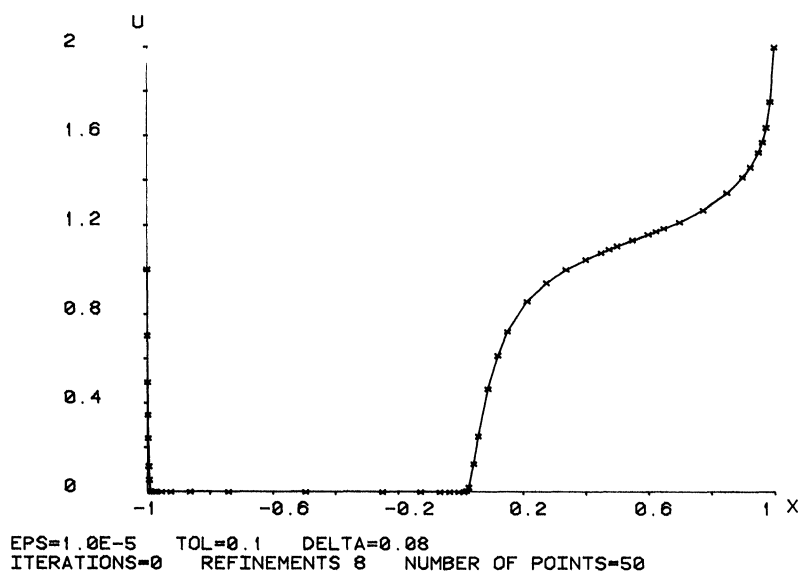


FIG. 6

$$\text{EPS} \cdot D^2Y/DX^2 = D(-X \cdot X \cdot Y)/DX + (2X-1)Y$$

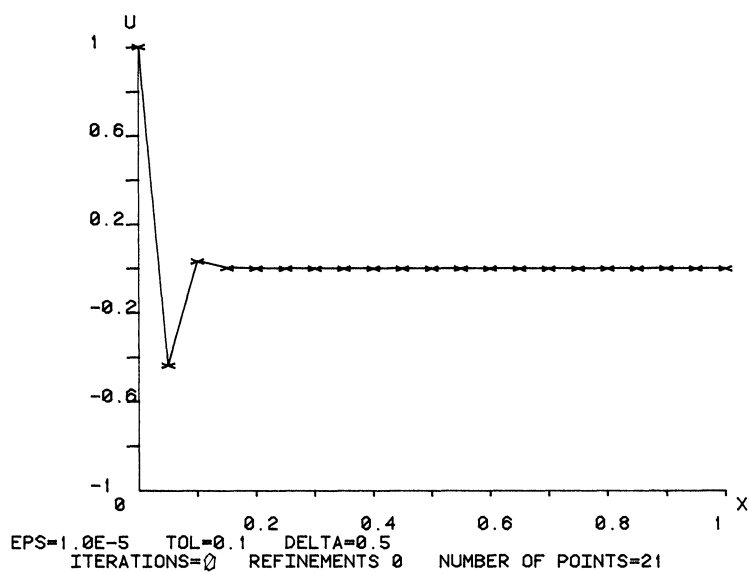


FIG. 7

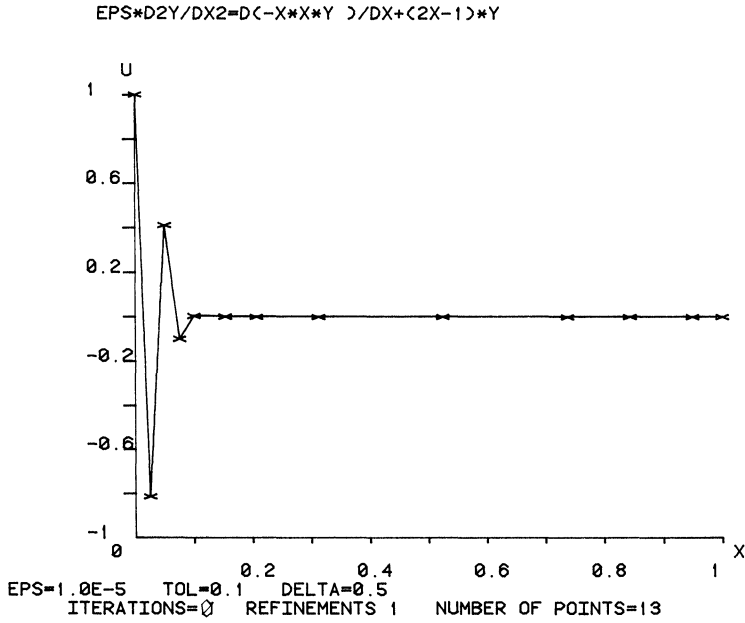


FIG. 8

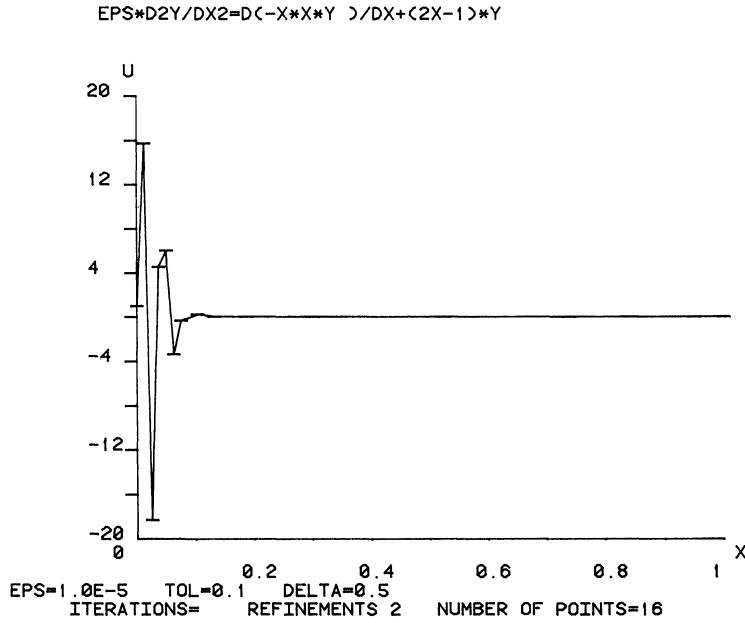


FIG. 9

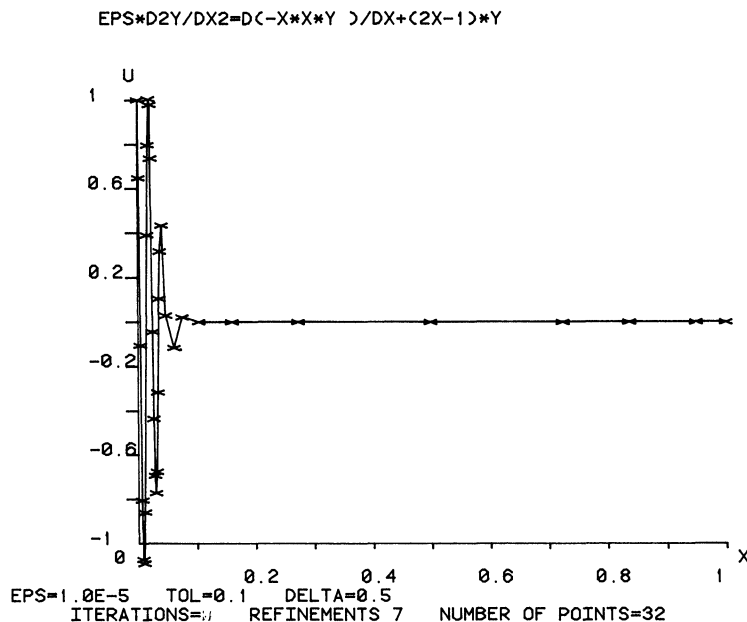


FIG. 10

6. Nonlinear equations. We consider now nonlinear equations

$$(6.1) \quad \begin{aligned} \varepsilon y'' &= (a(x, y))' + b(x, y), & -c \leq x \leq d, \\ y(-c) &= \alpha, & y(d) = \beta, \end{aligned}$$

and rewrite them as first-order systems

$$(6.2) \quad \begin{aligned} \varepsilon y' &= a(x, y) + v, \\ v' &= b(x, y). \end{aligned}$$

We use Newton's method to solve the system. Let $y^{(n)}, v^{(n)}$ be an approximation to the solution of (6.2). Then we linearize (6.2) around $y^{(n)}, v^{(n)}$ to obtain new approximations

$$y^{(n+1)} = y^{(n)} + \tilde{y}, \quad v^{(n+1)} = v^{(n)} + \tilde{v},$$

where u, v are solutions of the linearized system

$$(6.3) \quad \begin{aligned} \varepsilon \tilde{y}' &= a_y(x, y^{(n)})\tilde{y} + \tilde{v} + f_1, & a_y = \frac{\partial a}{\partial y}, \\ \tilde{v}' &= b_y(x, y^{(n)})\tilde{y} + f_2, \end{aligned}$$

with

$$f_1 = -\varepsilon (y^{(n)})' + a(x, y^{(n)}) + v^{(n)}, \quad f_2 = -(v^{(n)})' + b(x, y^{(n)}).$$

The linear system (6.3) is solved by our method.

As an example, we have considered the equation

$$\begin{aligned} \varepsilon y'' &= -\frac{1}{2}(y^2)' + y, & -1 \leq x \leq 1, \\ y(-1) &= -1, & y(+1) = 2, \end{aligned}$$

and solved it for

$$\varepsilon = 0.1, 0.05, 0.02, 0.01, 0.005, 0.02, 0.001.$$

The initial guess was a straight line between $y = -1$ and $y = 2$. We used "the method of continuation"; i.e., we used the computed solution as an initial guess to solve the problem for the next ε . Figs. 11–14 show the printouts for $\varepsilon = 10^{-1}$, 10^{-2} , 10^{-3} . The sudden increase in the number of Newton iterations is due to the decrease of TOL from 10^{-2} to 10^{-3} .

$$\text{EPS} * D^2 Y / DX^2 = D(-Y * Y / 2) / DX + Y$$

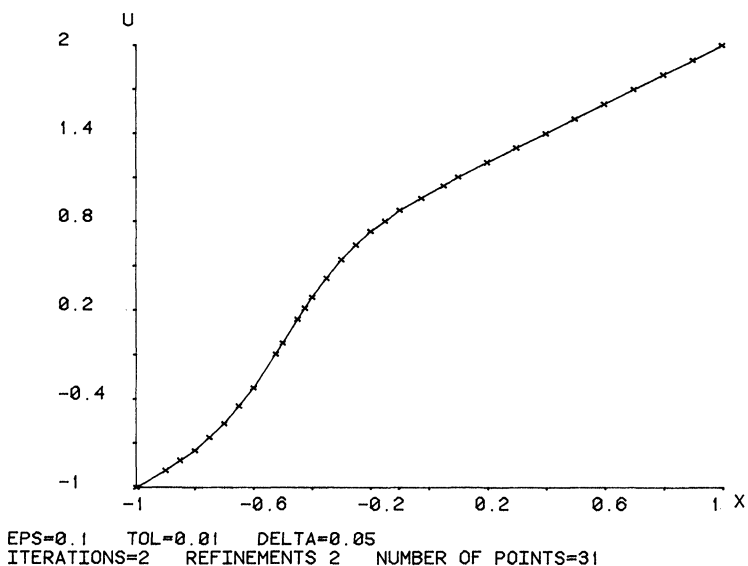


FIG. 11

$$\text{EPS} * D^2 Y / DX^2 = D(-Y * Y / 2) / DX + Y$$

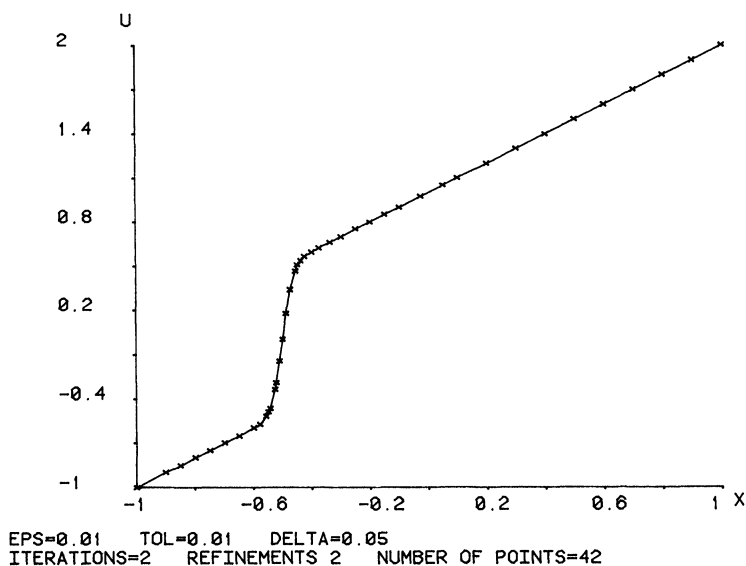


FIG. 12

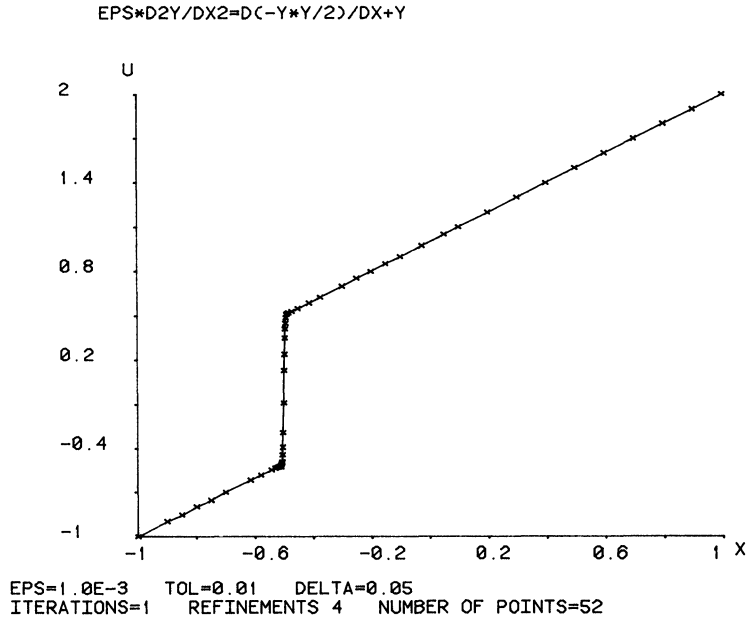


FIG. 13

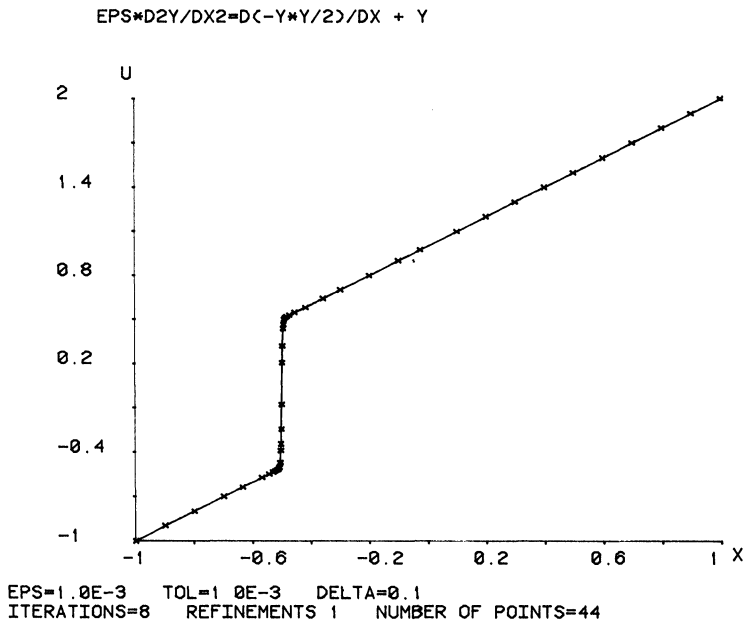


FIG. 14

REFERENCES

- [1] H. O. KREISS, *Numerical Methods for Singular Perturbation Problems*, SIAM-AMS Proceedings, vol. 10, American Mathematical Society, Providence RI, 1976.
- [2] H. O. KREISS AND N. NICHOLS, *Numerical methods for singular perturbation problems*, Report No. 57, Dept. of Computer Sciences, Uppsala University, 1975.
- [3] H. O. KREISS AND N. NICHOLS, to appear.
- [4] W. L. MIRANKER AND J. P. MORREEUW, *Semianalytic numerical studies of turning points arising in stiff boundary value problems*, Math. Comput., 28(1974), pp. 1017–1034.
- [5] C. E. PEARSON, *On a differential equation of boundary layer type*, J. Math. Phys., 47(1968), pp. 134–154.