Applied Mathematics & Information Sciences **4**(1) (2010), 31–45
– An International Journal

A New Algorithm for Solving Nonlinear Stationary Shock Problems with Mechanization

H. M. Habib[1] and E. R. El-Zahar[2]

[1]Department of Mathematics, King Saud University, El-Kharj, Kharj, Saudi Arabia
[2]Department of Mathematics, El-Menoufia University, School of Engineering
Shebin El-Kom, Egypt
Email Address: essam_zahar@hotmail.com

Received December 16, 2007; Revised January 9, 2008

Using the methods of mathematical analysis and computer algebra, we developed a new reliable algorithm of initial-value method for solving nonlinear singular perturbation problems. To mechanize the algorithm a MatLab procedure called *IVPHZ* was established. The *IVPHZ* can handle problems with one or two boundary layers as well as problems with an internal shock layer. Testing of the *IVPHZ* was made by solving numerous linear and nonlinear problems to demonstrate the implementation and efficiency of the procedure.

Keywords: Singular perturbation (SPPs), one-end boundary layer, internal layer, initial value method.

1 Introduction

Singular perturbation problems (SPPs) appear in many branches of applied mathematics [1] such as fluid dynamics, elasticity, optimal control, chemical reactor theory etc. The presence of the perturbation parameter leads to difficulties when numerical techniques are used to solve such problems. So SPPs have received a significant amount of attention in the past and recently. Some valid methods for solving these problems have been developed recently [2]– [22]. The difficulty in solving such problems lies on the huge amount of calculations involved and the high complexity in practice. Fortunately the rapid development of computer algebra systems can handle these difficulties besides its impact on the mathematical research [18]– [22]. Recently Wang *et al.* [21]– [23], Chen *et al.* [20], Zhenqing Li *et al.* [19], and Habib and El-Zahar [18] presented algorithms for solving many mathematical problems. These algorithms depend upon the progress of mathematical theories and computer algebra. The authors [18, 19] presented Matlab and Maple algorithms, respectively, to obtain the solution of linear SPPs with one boundary shock layer in different methods. The objective of this paper is to establish a promising algorithm which can solve nonlinear SPPs with one or two boundary shock layers as well as problems with an internal shock layer.

2 Initial Value Method

Consider the nonlinear two-point singularly-perturbed boundary-value problem

$$\varepsilon y'' + p(x,y)y' - q(x,y) = 0, \qquad x \in I = [0,1]\,, \tag{2.1}$$

with conditions $y(0) = A$ and $y(1) = B$. Here ε is a small positive parameter $(0 < \varepsilon \ll 1)$, $A$and B are given constants, $p(x,y)$ and $q(x,y)$ are assumed to be sufficiently continuously differentiable functions satisfying the following conditions:

$$p(x,y),\ q(x,y)\ \in C^2\left([0,1],\mathbb{R}\right)\ , \tag{2.2}$$

$$p_x(x,y) + q_y(x,y) \geq -\beta^2/4\varepsilon \quad x \in I, \quad y \in \mathbb{R}\,, \tag{2.3}$$

where $p(x,y) \geq \beta > 0$.

These conditions guarantee the existence of a unique solution $y \in C^4[0,1]$ of (2.1), see [4]. In general (2.1) has a boundary shock layer near the origin, see [4,5]. Also under these assumptions the reduced problem (corresponding to $\varepsilon = 0$ in Eq. (2.1)) becomes

$$p(x,u)\ u' - q(x,u) = 0, \qquad u(1) = B\,, \tag{2.4}$$

which has a unique solution $u \in C^3[0,1]$.

Equation (2.1) is given globally, i.e., for $y \in \mathbb{R}$. If we can find a priori bounded domain to which y belongs, then Eq. (2.1) can be restricted to that domain.

Equation (2.1) can be rewritten in the conservative form

$$\varepsilon y'' +\ f(x,y)' - g(x,y) = 0, \qquad x \in [0,1]\,, \tag{2.5}$$

with

$$f_y(x,y) = p(x,y)\ , \quad g(x,y) = q(x,y) + f_x(x,y)\,. \tag{2.6}$$

Theorem 2.1. *The singular perturbation problem (2.1) can be converted to the following alternative approximate forms:*

$$\begin{cases} f(x,u)\dfrac{du}{dx} - g(x,u)u = 0, & u(1) = B, \\ \varepsilon\dfrac{dy}{dx} + f(x,y) = f(x,u), & y(0) = A, \end{cases} \tag{2.7}$$

or in the form of layer correction problem

$$\begin{cases} f(x,u)\dfrac{du}{dx} - g(x,u)u = 0, & u(0) = B, \\ \dfrac{dv}{dt} = f(0,u(0)) - f(0, v + u(0)), & v(0) = A - u(0), \end{cases} \tag{2.8}$$

where $t = x/\varepsilon$ *and* $y(x) = u(x) + v(t)$.

Proof. Equation (2.1) can be rewritten in the conservative form (2.5). If $u(x)$ is the solution of the reduced problem (2.4), we can simply replace y by u in the $g(x,y)$ of Eq. (2.5) to obtain the approximate equation

$$\varepsilon y'' + f'(x,y) - g(x,u) = 0, \qquad x \in [0,1]. \tag{2.9}$$

Integrating Eq. (2.9) w.r.t. x we get an approximate equation with order ε [6].

$$\varepsilon y' + f(x,y) = \int^x g(t,u(t))\,dt + K, \tag{2.10}$$

where K is the constant of integration.

Substituting Eq. (2.4) and Eq. (2.6) into Eq. (2.10) we get

$$\begin{aligned} \varepsilon y' + f(x,y) &= \int^x \left[q(t,u(t)) + f_t(t,u(t))\right]\,dt + K \\ &= \int^x \left[p(y,u(t))\frac{du(t)}{dt} + f_t(t,u(t))\right]\,dt + K \\ &= \int^x \left[f_u(t,u(t))\frac{du(t)}{dt} + f_t(t,u(t))\right]\,dt + K \\ &= \int^x \frac{df(t,u(t))}{dt}\,dt + K \end{aligned}$$

or

$$\varepsilon y' + f(x,y) = f(x,u) + K. \tag{2.11}$$

In order to determine K we impose the condition that the reduced equation of (2.11) should satisfy the boundary condition $y(1) = B$. i.e.,

$$f(1,B) = f(1,B) + K.$$

Thus $K = 0$. Setting

$$y(x) = u(x) + v(t), \qquad t = x/\varepsilon,$$

in Eq. (2.11), we obtain the boundary layer corrected equation

$$\frac{dv}{dt} = f(\varepsilon t, u(\varepsilon t)) - f(\varepsilon t, u(\varepsilon t) + v(t)), \tag{2.12}$$

with the boundary condition $u(0) + v(0) = A$.

Letting $\varepsilon = 0$ we have

$$\frac{dv}{dt} = f(0, u(0)) - f(0, v(t) + u(0)) \tag{2.13}$$

with boundary condition $v(0) = A - u(0)$.

Obviously Eq (2.13) has a unique solution $v = v(t)$ with boundary layer behavior. Solving system (2.7) or (2.8) we get the approximate analytical solution of (2.1) as function of x *and* ε. □

Remark 2.2. The algorithm presented in [19] in the environment of Maple replaces the problem (2.1) in a special form (linear) with three equivalent (IVPs). In this work we get an equivalent (IVP) to the same problem (2.1) in the general form (quasilinear). Moreover, the proposed method can be used to solve problems with two boundary layers in addition to problems with internal layers which cannot be handled by the algorithm in [19].

3 A New Algorithm for Solving Nonlinear Stationary Shock Problems with Mechanization

Doing mathematics by computer is typical in the field of mathematics [7,8]. MatLab is a good mathematical environment to solve system (2.7) or (2.8) because of its great ability to perform both symbolic and numerical operations as well as manipulate graphics. Its powerful function library provides scientific calculation and programming with a friendly platform. Note that system (2.7) has independent initial conditions while system (2.8) has dependent ones. Hence system (2.7) can be solved directly using one statement in Matlab.

```
[u, y] = dsolve ('f (x, u)*Du-g(x, u)*u', 'epsilon*Dy + f(x, y)*y = f(x, u) u', 'u(1) =B',
'y(0) =A', 'x');
```

If $f(x, y)$ is linear or not highly nonlinear, or $p(x, y) = p(x)$, system (2.7) is simplified to [9]

$$\begin{cases} p(x)\dfrac{du}{dx} - q(x, u)u = 0, & u(1) = B, \\ \varepsilon\dfrac{dy}{dx} + p(x)y = p(x)u, & y(0) = A, \end{cases} \tag{3.1}$$

which can be solved using one statement in Matlab:

```
[u, y] = dsolve ('p (x)*Du-q(x, u)*u', 'epsilon*Dy + p(x)*y = p(x) u', 'u(1) =B', 'y(0)=A',
'x');
```

Note that the layer correction problem form (2.8) is more flexible for developing a general algorithm for solving problem (2.1) than (2.7). The main algorithm developed next called *IVPHZ* is based on system (2.8).

```
Function [Ysol, u, v] = IVPHZ (p, q, ab, AB, N, epsl)
```

```
if nargin < 4, error(’IVPHZ requires at least FOUR input arguments’); end
a=ab(1); b=ab(2); A=AB(1); B=AB(2);
syms x y v ep Dy Dv; clc
X0=x*(b-a) +a; % coordinate transformation from a domain [a, b] to the domain [0, 1]
p= (b-a)*subs (p,’x’, x0); q= (b-a) ˆ2*subs (q,’x’, x0); % coordinate transformation of p and q
s1=simplify(x/x*q/p);
sys1=str2mat (Dy-s1); iv1=str2mat (subs (’y (1) =B’,’B’, B));
If s1==0; u=B; else; u=simplify (dsolve (sys1, iv1,’x’)); end;
u0=subs (u,’x’,’0’);
f=subs (int (x*p/x, ’y’),’x’,’0’);
Fl=subs (f,’y’, u0+v);
Fr=subs (f,’y’, u0);
s2=simplify (Fl-Fr);
sys2=str2mat (Dv+s2); iv2=str2mat (subs (’v (0) =k’,’k’, A-u0));
v=simplify (dsolve (sys2, iv2,’t’));
v=subs (v,’t’, (x-a)/ (b-a)/epsilon);
Ysol = simplify (subs (u, ’x’, (x-a)/ (b-a)) +v);
If nargin =6; Y=subs (Ysol, ’x’, [a :( b-a)/N: b]);
Y=subs(Y, ’epsl’, ep); Plot ([a :( b-a)/N: b], Y);
else; error(’analytical output solution recommended with no graph’); end
```

IVPHZ is stored as an M-file in Matlab environment. Upon calling *IVPHZ* from the command line, Matlab passes the function and stores it in memory to be executed. This can be done by the following one command line:

[Y, u, v] =IVPHZ (p, q, [a, b], [A, B], N, epsl),

where

$\mathrm{p} = p(x,y)$, $\mathrm{q} = q(x,y)$,;
$[\mathrm{a}, \mathrm{b}]$: the end-points of the problem domain $[a, b]$ where we always consider the shock layer at $x = \mathrm{a}$ and b is the other end point;
$[\mathrm{A}, \mathrm{B}\,]$: the boundary conditions $y(a) = \mathrm{A}$, $y(b) = \mathrm{B}$;
u: the analytical solution of the reduced problem (at $\varepsilon = 0$);
v: the analytical solution of the correction problem;
Y: the analytical solution of the original problem (2.1) where $\mathrm{Y} = \mathrm{u} + \mathrm{v}$;
N: the number of grid points at which the graph of the output solution is plotted;
epsl: the ε value at which the graph of the output solution is plotted.

Note that *IVPHZ* does not plot the output solution if the values of N or ε are not given. The following message appears: “analytical output solution recommended with no graph”.

Moreover, if the input data to the algorithm is not complete, the program gives the error message: " 'IVPHZ requires at least FOUR input arguments' ".

The following examples illustrate the mechanized process and efficiency of the algorithm.

4 Examples with One Boundary Shock

Example 4.1. Consider the variable coefficient SPP [1]

$$\begin{gathered}\varepsilon y''(x) + \left(1 - \frac{x}{2}\right) y'(x) - \frac{1}{2} y(x) = 0, \qquad x \in [0,1], \\ y(0) = 0 \text{ and } y(1) = 1. \end{gathered} \tag{4.1}$$

A uniformly valid approximation is given by Nayfeh [10]

$$y(x) = \frac{1}{2-x} - \frac{1}{2} e^{-(x-x^2/4)/\varepsilon}.$$

The solution of this problem using *IVPHZ* is an easy matter. The user should input an one-line command:

```
[Y, u, v] = IVPHZ ('1-x/2', 'y/2', [0, 1], [0, 1], 1000, 1e-3).
```

The output is

$$y(x) = \frac{1}{2-x} - \frac{1}{2} e^{-x/\varepsilon}, \quad u(x) = \frac{1}{2-x}, \quad v(x) = -\frac{1}{2} e^{-x/\varepsilon}.$$

Since $p(x, y) = p(x)$, system (2.7) is reduced to (3.1) and the problem can be solved using the MatLab line command

```
[u,Y] = dsolve ('(1-x/2)*Du-1/2*u', 'e*Dy+ (1-x/2)*y-1/2','u(1) =1','y(0) =0','x').
```

The output results are

$$u(x) = \frac{1}{2-x}, y(x) = 0.5\sqrt{\pi/\varepsilon}\ e^{(x^2-4x+4)/2\varepsilon} \left(erf\left(\frac{x-2}{2\sqrt{\varepsilon}}\right) + erf\left(\frac{1}{\sqrt{\varepsilon}}\right)\right).$$

It is clear that the results obtained using the *IVPHZ* algorithm are simpler formulas than those obtained through system (2.7).

Example 4.2. Consider the linear SPP with boundary shock to the right given by [10]

$$\begin{gathered}\varepsilon y''(x) - y'(x) = 0 \qquad x \in [0,1], \\ y(0) = 1 \text{ and } y(1) = 0. \end{gathered} \tag{4.2}$$

The exact solution is given by

$$y(x) = \frac{(e^{(x-1)/\varepsilon} - 1)}{(e^{-1/\varepsilon} - 1)}.$$

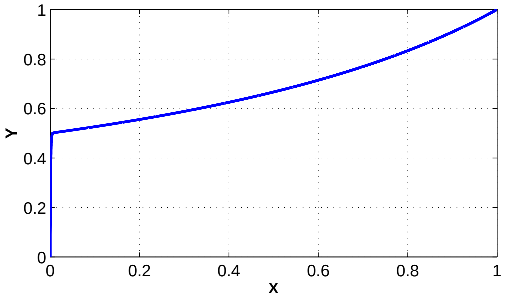

Figure 4.1: Approximate solution of example 4.1 at $\mathrm{N} = 1000, \ \ \varepsilon = 10^{-3}$using *IVPHZ* algorithm

Since the boundary shock is at $x = 1$, we should input the boundary condition in the reverse order form,

$\mathrm{ab} \ = [1, 0] \ \ , \mathrm{AB} = [\mathrm{y}(1), \mathrm{y}(0)]$
[Y, u, v] = IVPHZ ('-1', '0', [1, 0], [0, 1]).

The output solution is

$$y(x) = 1 - e^{-(1-x)/\varepsilon}.$$

Example 4.3. Consider the nonlinear SPP [11] given by

$$\begin{aligned} &\varepsilon y''(x) + 2y'^{y(x)} = 0 \qquad x \in [0, 1], \\ &y(0) = 0 \text{ and } y(1) = 0. \end{aligned} \tag{4.3}$$

A uniformly valid approximation is given by

$$y(x) = \log_e(2/(1 + x)) - (\log_e 2)e^{-2x/\varepsilon}.$$

Use the following command to invoke *IVPHZ*:

[u, v] = IVPHZ ('2', '-exp(y)', [0, 1], [0, 0]).

The output solution is

$$y(x) = \log_e \left(\frac{2}{x + 1} \right) - \log_e(2)e^{-2x/\varepsilon}.$$

Example 4.4. Consider the nonlinear SPP [5] given by

$$\begin{aligned} &\varepsilon y''(x) - y(x)y'(x) = 0 \qquad x \in \ [-1, 1], \\ &y(-1) = 0 \text{ and } y(1) = -1. \end{aligned} \tag{4.4}$$

A uniformly valid approximation is given by

$$y(x) = -(1 - e^{-(x+1)/\varepsilon})/(1 + e^{-(x+1)/\varepsilon}).$$

Use the following command to invoke *IVPHZ*:

```
[Y, u, v] = IVPHZ ('-y', '0', [-1, 1], [0,-1]).
```

The output solution is

$$y(x) = -1 + \frac{2}{1 + e^{(x+1)/\varepsilon}},$$

or simply

$$y(x) = -\frac{1 - e^{-(x+1)/\varepsilon}}{1 + e^{-(x+1)/\varepsilon}}.$$

Example 4.5. Consider the nonlinear SPP [12] given by

$$\begin{gathered} \varepsilon y''^{y(x)} y'(x) - \frac{\pi}{2}\sin(\pi x/2)e^{2y(x)} = 0 \qquad x \in [0,1], \\ y(0) = 0 \text{ and } y(1) = 0. \end{gathered} \tag{4.5}$$

A uniformly valid approximation is given by

$$y(x) = -\log_e \left[(1 + \cos(\pi x/2))(1 - 0.5e^{-x/2\varepsilon})\right].$$

Use the following command to invoke *IVPHZ*:

```
[Y, u, v] = IVPHZ ('exp(y)', 'pi/2*sin (pi*x/2)*exp (2*y)', [0, 1], [0, 0]).
```

The output solution is

$$y(x) = -\log_e(1 + cos(\pi x/2)) - \log\left[-1 + 2e^{x/2\varepsilon}\right] + \frac{x}{2\varepsilon} + \log(2).$$

Example 4.6. Consider the nonlinear SPP [13] given by

$$\begin{gathered} \varepsilon y''(x) + y'(x) + (y(x))^2 = 0, \qquad x \in [0,1], \\ y(0) = 0 \text{ and } y(1) = 0.5. \end{gathered} \tag{4.6}$$

A two-term asymptotic approximation was obtained [13] for comparison

$$y(x) = \frac{1}{1+x} - (1+x)^2 e^{-x/\varepsilon}.$$

Input the command

```
[Y, u, v] = IVPHZ ('1', '-y^2', [0, 1], [0, 1/2]).
```

The output solution is

$$y(x) = \frac{1}{x+1} - e^{-x/\varepsilon}.$$

Example 4.7. Consider the nonlinear SPP [13] given by

$$\begin{gathered} \varepsilon y''^2 y)y'^2 = 0 \qquad x \in [0,1], \\ y(0) = 1 \text{ and } y(1) = 0. \end{gathered} \tag{4.7}$$

A uniformly valid approximation is given by

$$y(x) = \frac{(1-2\varepsilon^2)e^{-x/\varepsilon}}{(x^2+2\varepsilon(x+\varepsilon)e^{-x/\varepsilon}+1}.$$

Input the command

```
[Y, u, v] = IVPHZ ('1-2*x^2*y', '2*x*y^2', [0, 1], [1, 0]).
```

The output solution is

$$y(x) = e^{-x/\varepsilon}.$$

Example 4.8. Consider the nonlinear SPP [13] given by

$$\varepsilon y'' + \left[\frac{x+2y+2xy}{(x+2)(x+1)^2}\right]y' + \left[\frac{(y+1)(2x^2y+x(5y-3)+3y-5}{(x+2)^2(x+1)^3}\right] = 0, \qquad x\in[0,1], \quad y(0)=0.5 \text{ and } y(1)=1. \tag{4.8}$$

The solution as an asymptotic expansion is given by [13]

$$y(x) = \frac{3-e^{-x/\varepsilon}}{3+e^{-x/\varepsilon}}.$$

Using the *IVPHZ* algorithm, the output solution is

$$y(x) = 1 - \frac{2}{1+3e^{x/\varepsilon}},$$

or simply

$$y(x) = \frac{3-e^{-x/\varepsilon}}{3+e^{-x/\varepsilon}}.$$

5 Examples with Two–Boundary Shock

The algorithm is extended to include problems with two boundary shock layers. Consider problem (2.1) with

$$p(x^*,y)=0, \qquad p_x(x^*,y)<0, \quad x^*\in(0,1)\,, \;\; x\in I, \quad y\in\mathbb{R}. \tag{5.1}$$

There is a turning point at $x = x^*$. In addition, under certain additional conditions [14] the solution $y(x)$ has an exponential boundary shock layer at each end.

Theorem 5.1. *The singular perturbation problem (2.1) with conditions (5.1) and with the aid of [14] can be converted to the approximate form:*

$$\begin{cases} f(x,u)\dfrac{du}{dx} - g(x,u)u = 0, & u(x^*)=y^*, \\ \varepsilon\dfrac{dy_\pm}{dx_\pm} + f(x_\pm,y_\pm) = f(x_\pm,u), & y_-(0)=A, \quad y_+(1)=B, \end{cases} \tag{5.2}$$

where

$$x_-\in I_-=[0,x^*], \qquad x_+\in I_+=[x^*,1], \qquad y_-=y(x_-), \qquad y_+=y(x_+).$$

Therefore the *IVPHZ* algorithm can be applied to turning-point problems with a two-boundary shock layers by considering the original problem as two SPPs, each with single boundary layer/shock.

Example 5.1. Consider the singular perturbation problem [15] given by

$$\begin{gathered}\varepsilon y'' - 2(2x-1)y' - 4y = 4(4x-1), \qquad x \in [0,1], \\ y(0) = 1 \text{ and } y(1) = 1.\end{gathered} \tag{5.3}$$

The solution to the problem is given by

$$y(x) = -2x + \exp(-2x/\varepsilon) + 3\exp(-2(1-x)/\varepsilon) + O(\varepsilon).$$

The problem has a turning point at $x^* = 1/2$ and $y(x^*) = 0$.

Since the shock layer is at $x = 0$, we should use $\mathrm{ab} = [0, \mathrm{x}^*]$, $\mathrm{AB} = [\mathrm{y}(0), \mathrm{y}(\mathrm{x}^*)]$, and the following command to obtain the solution $y_-(x_-)$, $x_- \in [0, 1/2]$:

```
[Y, u, v] = IVPHZ ('-2*(2*x-1)', '4*y+4*(4*x-1)', [0, 1/2], [1, 0]).
```

The output results are

$$y_- = -2x + e^{-2x_-/\varepsilon}, \quad u(x) = -2x, \quad v_- = e^{-2x_-/\varepsilon}.$$

Since the shock layer is at $x = 1$, we should use $\mathrm{ab} = [1, \mathrm{x}^*]$, $\mathrm{AB} = [\mathrm{y}(1), \mathrm{y}(\mathrm{x}^*)]$ and the next command to obtain the solution $y_+(x_+)$, $x_+ \in [1/2, 1]$.

```
[Y, u, v] = IVPHZ ('-2*(2*x-1)', '4*y+4*(4*x-1)', [1, 1/2], [1, 0]).
```

The output results are

$$y_+ = -2x + 3e^{-2(1-x_+)/\varepsilon}, \quad u(x) = -2x, \quad v_+ = 3e^{-2(1-x_+)/\varepsilon}.$$

Example 5.2. Consider the nonlinear singular perturbation problem [14] given by

$$\begin{gathered}\varepsilon y''^2 + 1)y'^2 + 1) = 0, \qquad x \in [-1,1], \\ y(-1) = 1 \text{ and } y(1) = 1.\end{gathered} \tag{5.4}$$

The asymptotic expansion solution is given by

$$y(x) = \frac{\sqrt{3}\exp((x^2-1)/(2\varepsilon))}{\sqrt{4 - \exp((x^2-1)/\varepsilon)}}.$$

The above solution can be simplified for comparison in the form

$$y(x) = \frac{\sqrt{3}}{\sqrt{-1 + 4\exp((1-x^2)/\varepsilon)}}.$$

The problem has a turning point at $x^* = 0$, and $y(x^*) = 0$.

Since the shock layer is at $x = -1$, we should input $\mathrm{ab} = [-1, \mathrm{x}^*]$, $\mathrm{AB} = [\mathrm{y}(-1), \mathrm{y}(\mathrm{x}^*)]$ and the following command to obtain the solution at $y_-(x_-)$, $x_- \in [-1, 0]$:

```
[Y, u, v] = IVPHZ ('-x*(y^2+1)', 'y*(y^2+1)', [-1, 0], [1, 0]).
```

The solution is

$$y_- = \frac{\sqrt{3}}{\sqrt{-1 + 4exp(2(1 + x_-)/\varepsilon)}}.$$

Since the shock layer is at $x = 1$, we should input $\mathrm{ab} = [1, \mathrm{x}^*]$, $\mathrm{AB} = [\mathrm{y}(1), \mathrm{y}(\mathrm{x}^*)]$ and use the following command to obtain the solution at $y_+(x_+)$, $x_+ \in [0, 1]$:

```
[Y, u, v] = IVPHZ ('-x*(y^2+1)', 'y*(y^2+1)', [1, 0], [1, 0]).
```

The solution is

$$y_+ = \frac{\sqrt{3}}{\sqrt{-1 + 4exp(2(1 - x_+)/\varepsilon)}}.$$

6 Examples with Interior Boundary Shock

The algorithm can be extended to solve internal layer/shock problems as well. Consider problem (2.1) with

$$p(x, y) = p(y), \quad p(y^*) = 0\,, \quad q_y(x, y) \geq q_* > 0, \qquad x \in I, \quad y \in \mathbb{R}. \tag{6.1}$$

Under certain additional conditions [16] the solution has a shock layer at $x = x^*$, where $x^* \in (0, 1)$.

For any $m = 0, 1, 2, \ldots$, we have

$$m + 1 \geq 1. \tag{6.2}$$

Theorem 6.1. *The singular perturbation problem (2.1) with conditions (6.1) and with the aid of [16] can be converted to the following form:*

$$\begin{cases} f(x_\pm, u_\pm)\dfrac{du_\pm}{dx_\pm} - g(x_\pm, u_\pm)u_\pm = 0, & u_-(0) = A,\ u_+(1) = B, \\ \varepsilon\dfrac{dy_\pm}{dx_\pm} + f(x_\pm, y_\pm) = f(x_\pm, u), & y_-(x^*) = y_+(x^*) = y^*, \end{cases} \tag{6.3}$$

where $x_- \in I_- = [0, x^*]$, $x_+ \in I_+ = [x^*, 1]$, $y_- = y(x_-), y_+ = y(x_+), u_- = u(x_-), u_+ = u(x_+)$.

Example 6.1. Consider the nonlinear singular perturbation problem [17] given by

$$\begin{aligned} \varepsilon y'' + yy' - y = 0, &\qquad x \in [0, 1], \\ y(0) = -1/2 \text{ and } y(1) = 1. & \end{aligned} \tag{6.4}$$

with at turning point at $x^* = 1/4$ and $y^* = 0$.

The problem has approximate solutions [17] of the form

$$\begin{aligned} y_+ &= u_+ - u_+ \left[1 + \tanh(-u_+(x_+ - 1/4)/2\varepsilon)\right], \\ y_- &= u_- - u_- \left[1 + \tanh(-u_-(x_- - 1/4)/2\varepsilon)\right]. \end{aligned}$$

Since the shock layer is located at $x^* = 1/4$, we should input $\mathrm{ab} = [\mathrm{x}^*, 0]$, $\mathrm{AB} = [\mathrm{y}(\mathrm{x}^*),\mathrm{y}(0)]$ and use the following command to obtain the solution at $y_-(x_-),\ \ x_- \in [0, 1/4]$:

[Y, u, v] = IVPHZ ('y', 'y', [1/4, 0], [0, -1/2]).

The results are

$$y_- = \frac{1}{2}\left(2x - 1 + \frac{1}{1 + e^{(-4x_- + 1)/16\varepsilon}}\right), \qquad u_-(x_-) = x_- - 1/2,$$

$$v_-(x_-) = \frac{1}{2 + 2e^{(-4x_- + 1)/16\varepsilon}}.$$

Since the shock layer is located at

$$x^* = 1/4,$$

we should input $\mathrm{ab} = [\mathrm{x}^*, 1],\ \ \mathrm{AB} = [\mathrm{y}(\mathrm{x}^*),\mathrm{y}(1)]$ and use the following command to obtain the solution $y_+(x_+),\ \ x_+ \in [1/4, 1]$:

[Y, u, v] = IVPHZ ('y', 'y', [1/4, 1], [0, 1]).

The output results are

$$y_+ = \left(x_+ + \frac{1}{1 + e^{(4x_+ - 1)/16\varepsilon}}\right), u_+(x_+) = x_+, v_+(x_+) = \frac{1}{2 + 2e^{(4x_+ - 1)/16\varepsilon}}.$$

Table 6.1 lists the maximum numerical errors of the obtained solutions using *IVPHZ* algorithm for different values of ε.

Table 6.1 indicates that the obtained approximate analytical solutions using *IVPHZ* approximates the given problems solution very well. In some cases the given solution of the problems were retrieved.

7 Conclusion

A new reliable algorithm, *IVPHZ*, based upon the initial-value method and designed for solving nonlinear singular perturbation problems with stationary shock is developed. The *IVPHZ* based on the deduced equivalent (IVP) can give the solution to SPP and to the reduced problem as well as to the layer correction problem in a single statement. *IVPHZ* not only gives the solution of SPP with one boundary layer but also the solution of turning-point

Table 6.1: Maximum error of the obtained approximate analytical solution using *IVPHZ* algorithm

Test Problem	Maximum Error at different values of ε for each test problem				
	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-7}$
4.1	6.7207e-005	6.7181e-006	6.7179e-007	6.7179e-008	6.7179e-009
4.2	0	0	0	0	0
4.3	2.2204e-016	2.2204e-016	2.2204e-016	2.2204e-016	2.2204e-016
4.4	0	0	0	0	0
4.5	1.8874e-015	1.8874e-015	1.8874e-015	1.8874e-015	1.8874e-015
4.6	7.3524e-004	7.3489e-005	7.3486e-006	7.3486e-007	7.3486e-007
4.7	8.8820e-008	8.8820e-010	8.8820e-012	8.8735e-014	8.3267e-016
4.8	0	0	0	0	0
5.1	0	0	0	0	0
5.2	2.3627e-004	2.3627e-005	2.3603e-006	2.3627e-007	2.3627e-008
6.1	2.1089e-003	2.1089e-004	2.1139e-005	2.0083e-006	2.1103e-008

problems exhibiting internal layer or two boundary layers. The results of the examples indicate that *IVPHZ* has many advantages including easy handling, efficiency, excellent property for operation and powerful competence.

References

[1] J. Kevorkian and J. D. Cole, *Perturbation Methods in Applied Mathematics*, Springer-Verlag, New York, 1981.

[2] M. K. Kadalbajoo and K. C. Patidar, A survey of numerical techniques for solving singularly perturbed ordinary differential equations, *Appl. Math. Comput.* **130** (2002), 457–510.

[3] M. K. Kadalbajoo and Y. N. Reddy, Asymptotic and numerical analysis of singular perturbation problems: a survey, *Appl. Math. Comput.* **30** (1989), 223–259.

[4] J. Lorenz, Stability and monotonicity properties of stiff quasilinear boundary problems, *Univ. u Novom Sadu, Zb. Rad. Prirod.-Mat. Fak. Ser. Mat.* **12** (1982), 151–175.

[5] R. E. O'Malley, *Introduction to Singular Perturbations*, Academic Press, New York and London, 1974.

[6] H. M. Habib and E. R. El-Zahar, Effective initial value method for nonlinear singular perturbation problems, unpublished work, submitted to the *Appl. Math. Comput.* (2007).

[7] A. Friendman, J. Glimm and J. Lavery, The Mathematical and Computational Sciences in Emerging Manufacturing Techniques and Management Practices, *SIAM* (1992), 62–63.

[8] J. G. Glimm, *Mathematical Sciences, Technology, and Economic Competitiveness*, National Academy Press, Washington, DC, 1992.

[9] H. M. Habib and E. R. El-Zahar, Variable step size initial value algorithm for singular perturbation problems using locally exact integration, *Appl. Math. Comput.* **200** (2008), 330–340.

[10] A. H. Nayfeh, *Perturbation Methods*, Wiley, New York, 1973.

[11] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1978.

[12] R. Vulanovi, A higher-order method for stationary shock problems, *Appl. Math. Comput.* **108** (2000), 139–152.

[13] R. S. Johnson, *Singular Perturbation Theory - Mathematical and Analytical Techniques with Application to Engineering*, Springer-Verlag, New York, 2005.

[14] P. Lin, A numerical method for quasilinear singular perturbation problems with turning points, *Computing* **46** (1991), 155–164.

[15] S. Natesan, J. Jayakumar and J. Vigo-Aguiar, Parameter, uniform numerical method for singularly perturbed turning point problems exhibiting boundary layer, *J. Comput. Appl. Math.* **158** (2003), 121–134.

[16] J. Lorenz, Analysis of difference schemes for a stationary shock problem, *SIAM J. Numerical Analysis* **21** (1984), 1038–1053.

[17] R. E. O'Malley, Shock and transition layer for singularly perturbed second-order vector systems, *SIAM J. Applied Mathematics* **43** (1983), 935–943.

[18] H. M. Habib and E. R. El-Zahar, An algorithm for solving singular perturbation problems with mechanization, *Appl. Math. Comput.* **188** (2007), 286–302.

[19] Zhenqing Li and Weiming Wang, Mechanization for solving SPP by reducing order method, *Appl. Math. Comput.* **169** (2005), 1028–1037.

[20] W. Chen and Z. Lu, An algorithm for a domain decomposition method, *Appl. Math. Comput.* **154** (2005), 221–235.

[21] W. Wang, A new algorithm for solving DAEs with mechanization, *Appl. Math. Comput.* **167** (2005), 1350–1372.

[22] W. Wang and X. Lian, A new algorithm for symbolic integral with application, *Appl. Math. Comput.* **162** (2005), 949–968.

[23] W. Wang and C. Lin, A new algorithm for integral of trigonometric functions with mechanization, *Appl. Math. Comput.* **164** (2005), 71–82.

Hamza Habib received both MS and Ph.D. from the department of Mechanical and Aerospace Engineering at Arizona State University in December 1984 and December 1989 respectively. During his work at ASU he developed with Dr. B. Wood and other colleagues the absorber for the open cycle of the Solar Refrigeration System. He worked as an assistant professor at the school of Engineering, Shebin El-Kom, Egypt between 1991-2006. In January 2007 he accepted a job at King Saud University, Saudi Arabia in the department of mathematics in Al-Kharj, later in November 2007 he was appointed as the acting chairman of the department. He has written several articles in heat transfer, biochemical engineering and Numerical methods.

Essam El Zahar received both MS and Ph.D. from the Department of Basic Engineering Sciences at the Faculty of Engineering, Shebin El-Kom, Menofia University, Egypt in July 2004 and August 2008 respectively. Now, He works as an assistant professor at the school of Engineering, Shebin El-Kom, He has written several articles in Numerical methods.