Numerical Solutions of Third-Order Boundary Value Problems

Jishan Ahmed
Mathematical Sciences Department
Ball State University
e-mail : jahmed@bsu.edu

Abstract. In this project, the aim is to approximate the numerical solutions of the third-order boundary value problems using finite difference method. The main feature of the finite difference method is to obtain discrete equations by replacing derivatives with appropriate finite differences. In this work, finite difference method for the third-order boundary value problems have been derived in four steps. In the first step, we have discretized the domain of the problem. In the second step, we have discretized the differential equation at the interior nodal points. The third step is devoted to the implement of the boundary conditions. Finally, the resulting linear systems have been solved using Gaussian elimination method.

1. Introduction

The numerical solution of third-order boundary value problems (BVPs) is of great importance due to its wide application in scientific research. The third-order differential equations arise in many physical problems such as electromagnetic waves, thin film flow, and gravity-driven flows [1, 2, 4, 5]. In this project, finite difference method (FDM) is used to obtain a numerical solution to the third-order boundary value problems of the following form:

$$u_{xxx}(x) = f(x) \tag{1.1}$$

with boundary conditions

$$u(a) = A_1, \quad u'(a) = A_2, \quad y(b) = A_3,$$

where $A_i (i = 1, 2, 3)$ are finite real constants.

2. Numerical Method and Formulation

Finite difference method [6] for the solution of a third-order boundary value problem consists in replacing the derivatives occurring in the differential equation

Key words and phrases: Finite difference method (FDM), Boundary value problems, Numerical solutions.

(and in the boundary conditions as well) by means of their finite-difference approximations. For this purpose, we discretize the domain $[a, b]$ into N equal parts: $[a, b] = [x_0, x_1] \cup [x_1, x_2] \cup \ldots \cup [x_{N-1}, x_N]$ where $a = x_0 < x_1 < \ldots < x_{N-1} < x_N = b$ are the grid points or nodal poits. Denote $h = (b-a)/N$, called the step size. Then the node points are given by $x_i = a + ih$, $0 \leq i \leq N$.

Nonuniform partition of the interval is also possible, and is in fact preferred if the the solution of the boundary value problems (1.1) changes much more rapidly in some part of the interval than the remaining parts. We have considered here uniform partition for domain discretization. Now, we will discretize the BVPs (1.1) at the interior node points $x_1, \ldots, x_{N-1}$. For this purpose, Taylor expansions are used. To derive an approximation for u_{xxx} at $x = x + h/2$, let

$$u_{xxx} \approx D_h^{(3)} u(x) \equiv \alpha u(x - \frac{3}{2}h) + \beta u(x - \frac{h}{2}) + \gamma u(x + \frac{h}{2}) + \delta u(x + \frac{3}{2}h) \tag{1.2}$$

The coefficients $\alpha, \beta, \gamma, \delta$ are to be determined in such a way that this linear combination is indeed an approximation of the third derivative. We have obtained the following results using Taylor polynomial [6] approximations

(1.3)
$$u(x-\frac{3}{2}h) \approx u(x)+\frac{(-\frac{3h}{2})}{1!}u'(x)+\frac{(-\frac{3h}{2})^2}{2!}u''(x)+\frac{(-\frac{3h}{2})^3}{3!}u'''(x)+\frac{(-\frac{3h}{2})^4}{4!}u''''(x)+\frac{(-\frac{3h}{2})^5}{5!}u'''''(x)+\ldots$$

(1.4)
$$u(x-\frac{h}{2}) \approx u(x)+\frac{(-\frac{h}{2})}{1!}u'(x)+\frac{(-\frac{h}{2})^2}{2!}u''(x)+\frac{(-\frac{h}{2})^3}{3!}u'''(x)+\frac{(-\frac{h}{2})^4}{4!}u''''(x)+\frac{(-\frac{h}{2})^5}{5!}u'''''(x)+\ldots$$

(1.5)
$$u(x+\frac{h}{2}) \approx u(x)+\frac{(\frac{h}{2})}{1!}u'(x)+\frac{(\frac{h}{2})^2}{2!}u''(x)+\frac{(\frac{h}{2})^3}{3!}u'''(x)+\frac{(\frac{h}{2})^4}{4!}u''''(x)+\frac{(\frac{h}{2})^5}{5!}u'''''(x)+\ldots$$

(1.6)
$$u(x+\frac{3}{2}h) \approx u(x)+\frac{(\frac{3h}{2})}{1!}u'(x)+\frac{(\frac{3h}{2})^2}{2!}u''(x)+\frac{(\frac{3h}{2})^3}{3!}u'''(x)+\frac{(\frac{3h}{2})^4}{4!}u''''(x)+\frac{(\frac{3h}{2})^5}{5!}u'''''(x)+\ldots$$

Substitute into (1.2) and rearrange:

$$
\begin{aligned}
D_h^{(3)}u(x) \approx (\alpha+\beta+\gamma+\delta)u(x) &+ \left\{\frac{(-\frac{3h}{2})}{1!} + \frac{(-\frac{h}{2})}{1!}\beta + \frac{(\frac{h}{2})}{1!}\gamma + \frac{(\frac{3h}{2})}{1!}\delta\right\}u'(x) \\
&+ \left\{\frac{(-\frac{3h}{2})^2}{2!}\alpha + \frac{(-\frac{h}{2})^2}{2!}\beta + \frac{(\frac{h}{2})^2}{2!}\gamma + \frac{(\frac{3h}{2})^2}{2!}\right\}u''(x)+ \\
&\left\{\frac{(-\frac{3h}{2})^3}{3!}\alpha + \frac{(-\frac{h}{2})^3}{3!}\beta + \frac{(\frac{h}{2})^3}{3!}\gamma + \frac{(\frac{3h}{2})^3}{3!}\delta\right\}u'''(x)+ \\
&\left\{\frac{(-\frac{3h}{2})^4}{4!}\alpha + \frac{(-\frac{h}{2})^4}{4!}\beta + +\frac{(\frac{h}{2})^4}{4!}\gamma + \frac{(\frac{3h}{2})^4}{4!}\right\}u''''(x) \\
&+ \left\{\frac{(-\frac{3h}{2})^5}{5!}\alpha + +\frac{(-\frac{h}{2})^5}{5!}\beta + \frac{(\frac{h}{2})^5}{5!}\gamma + \frac{(\frac{3h}{2})^5}{5!}\delta\right\}u'''''(x) + \ldots
\end{aligned}
$$

To have

$$D_h^{(3)} \approx u\prime\prime\prime(x) \tag{1.7}$$

for arbitrary functions $u(x)$, we have equated the coefficients of $u(x), u'(x), u''(x)$ and $u'''(x)$ on both sides and obtain the following linear system:

$$
\begin{aligned}
\alpha+\beta+\gamma+\delta &= 0 && (1.8)\\
-3\alpha-\beta+\gamma+3\delta &= 0 && (1.9)\\
9\alpha+\beta+\gamma+9\delta &= 0 && (1.10)\\
-27\alpha-\beta+\gamma+27\delta &= \frac{48}{h^3} && (1.11)
\end{aligned}
$$

The above system has the unique solution:

$\alpha = -\frac{1}{h^3}, \quad \beta = \frac{3}{h^3}, \quad \gamma = -\frac{-3}{h^3}, \quad \delta = -\frac{1}{h^3}.$

This determines

$$D_h^{(3)}u(x) \approx -\frac{1}{h^3}u(x-\frac{3}{2}h) + \frac{3}{h^3}u(x-\frac{h}{2}) - \frac{3}{h^3}u(x+\frac{h}{2}) - \frac{1}{h^3}u(x+\frac{3}{2}h) \tag{1.12}$$

For the error, we have,

$$D_h^{(3)}u(x) \approx u'''(x) + \frac{1}{8}h^2u'''''(x) \tag{1.13}$$

and so

$$u'''(x) - D_h^{(3)}u(x) \approx -\frac{1}{8}h^2u'''''(x) \tag{1.14}$$

Finally we have obtained the following finite difference scheme at $x = x_{i+1/2}$ becomes

$$\frac{-u_{i-2} + 3u_{i-1} - 3u_i + u_{i+1}}{h^3} = f(x_i - h/2), 2 \leq i \leq N-1 \tag{1.15}$$

Now we need two more equations and they are formulated from discretization of the boundary conditions. For the model problem (1.1), the discretization of the boundary condition is straight forward:
$u(a) = A_1, \quad u_{(a+h)} = u(a) + h * A_2 \quad$ and $u(b) = A_3$ Therefore, the difference system for the unknown numerical solution vector

$$\overline{u} = [u_1, \ldots, u_N - 1]^T$$

is

$$A\overline{u} = \overline{b} \tag{1.16}$$

where

$$A = \begin{pmatrix} -3 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 3 & -3 & 1 & \ddots & & & & \vdots \\ -1 & 3 & -3 & 1 & \ddots & & & \vdots \\ 0 & -1 & 3 & -3 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -1 & 3 & -3 & 1 & 0 \\ \vdots & & & \ddots & -1 & 3 & -3 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 & -1 & 3 & -3 \end{pmatrix} \tag{1.17}$$

is the coefficient matrix and

$$\overline{b} = [h^3 f(a + 2h - h/2), \ldots, h^3 f(a + Nh - h/2)]^T$$

is the right hand side vector.The above linear system has the truncation error of order $O(h^2)$.

3. Numerical Results

Firstly, we have tested our Matlab code for a known boundary value problem. Then we have implemented our code for our proposed boundary value problem. All calculations are performed by $MATLAB \quad R2013b$

Test Example:

Consider the boundary value problem:

$$u^{(3)}(x) = 24x \tag{1.18}$$

subject to the boundary conditions

$$u(0) = 0, \quad u'(0) = -1, \quad u(1) = 0.$$

The analytical solution of this problem is

$$u(x) = x^4 - x. \tag{1.19}$$

The following *MATLAB* code implements the finite difference (1.16) for solving (1.18):

Matlab Main Function:

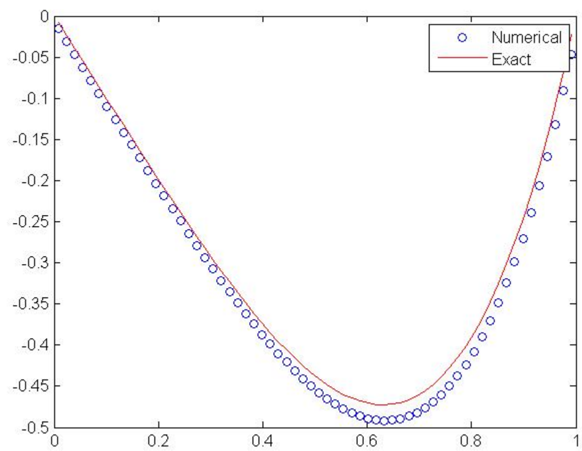
```
function [x,U] =fdm_trial(a,b,n)

h=(b-a)/n;
 A=sparse(n,n);
 F=zeros(n,1);
 %{ (-u(i-1)+3u(i)-3u(i+1)+u(i+2))/h^3=f(i), u_0=0,u_n=-1}
  for i =1:n
      A(i,i)=-3;
  end
  for i =1:n-1
      A(i+1,i)=3;
  end

  for i =1:n-2
      A(i+2,i)=-1;
  end

for i =1:n-1
      A(i,i+1)=1;
  end

  for j=1:n
      x(j)=a+j*h-h/2;
  end
  for j=1:n
      F(j)=(h^3*24*x(j));
  end
```

```

F(1) = F(1) + h;

%A(1,1) = -2;
size(A);
size(F);
% disp([F]);
m = size(A, 1);  % Size of input matrix
r = zeros(m, 1); % Imitialize permutatiom vector
for i = 1 : 1 : m
   r(i) = i;
end
% disp(r);
U = zeros(m, 1);
for k = 1 : 1 : m
    max = abs(A(r(k), r(k)));
    max_pos = k;
    for l = k : 1 : m
        if abs(A(r(l), r(k))) > max
            max = abs(A(r(l), r(k)));
            max_pos = l;
        end
    end
    temp_r = r;
    r(k) = temp_r(max_pos);
    r(max_pos) = temp_r(k);
    for i = 1 : 1 : m
        if i ~= k
            zeta = A(r(i), k) / A(r(k), k);
            for j = k : 1 : m
                A(r(i), j) = (A(r(i), j) - A(r(k), j) * zeta);
            end
            F(r(i)) = F(r(i)) - F(r(k)) * zeta;
        end
    end
end
for i = 1 : 1 : m
    U(i) = F(r(i)) / A(r(i), i);
end
end
```

Matlab Driver Program:

```
[X, U] = fdm_trial(0,1,64)
plot(X,U)
hold on
f = @(x) x.^4 - x;
%plot(X(1:50:end),f(X(1:50:end)),'o')
plot(X,f(X),'o')
```

Figure 1: Showing the comparison between approximate FDM solutions and exact solutions for test example when $n = 64$



We have tested our code for solving boundary value problems (1.18). We have observed from Figure (1)- (5) that there is a dramatic improvement in the solution for decreasing values of h. The approximate FDM solution with $h = 1/1024$ is much more accurate than the approximate solution with $h = 1/64$. It is also noticeable that smaller h requires more time. In our case, we have calculated that elapsed time with $h = 1/1024$ was 3351.457248 seconds. On the other hand, elapsed time with $h = 1/64$ was 1.126665 seconds.

Proposed Example:

Consider the boundary value problem:

$$u_{xxx} = f(x) \tag{1.20}$$

subject to the boundary conditions

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

As the right-hand side use $f(x)$ which is zero everywhere except it is 1 at $x = (a + b)/2$. We have tested our code for solving boundary value problems (1.20). We have observed from Figure (6)- (10) that there is a good improvement in the smoothness of the solution curve for decreasing values of h. The approximate FDM solution with $h = 1/1024$ is more smooth than the approximate solution with $h = 1/64$. It is also noticeable that smaller h requires more time. In our case,

Figure 2: Showing the comparison between approximate FDM solutions and exact solutions for test example when $n = 128$

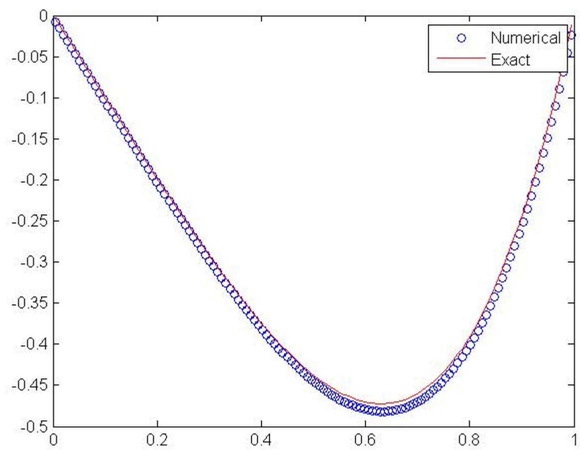


Figure 3: Showing the comparison between approximate FDM solutions and exact solutions for test example when $n = 256$

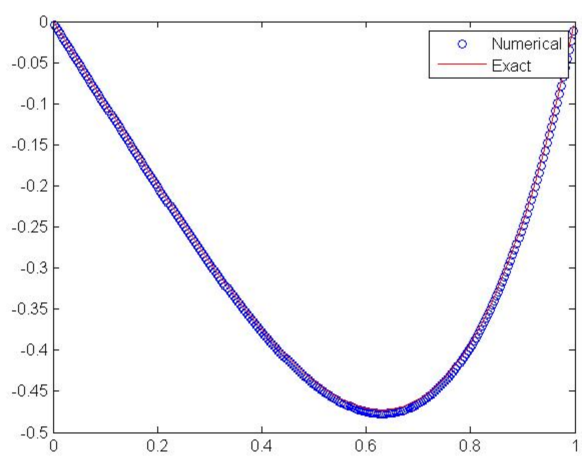


Figure 4: Showing the comparison between approximate FDM solutions and exact solutions for test example when $n = 512$

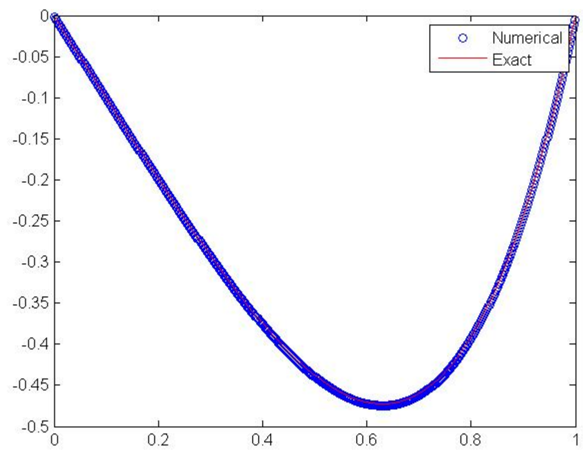


Figure 5: Showing the comparison between approximate FDM solutions and exact solutions for test example when $n = 1024$

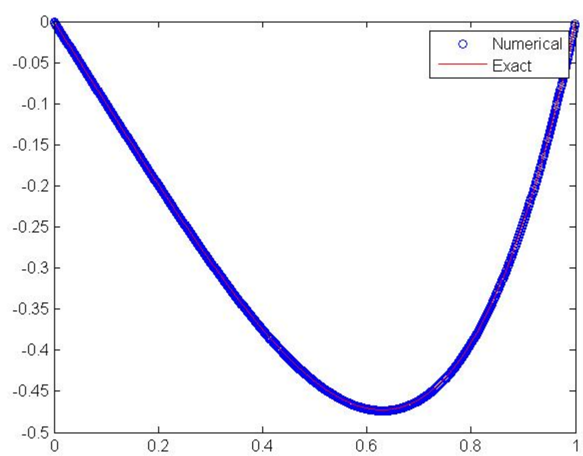


Figure 6: Showing the comparison between approximate FDM solutions and exact solutions for proposed example when $n = 64$

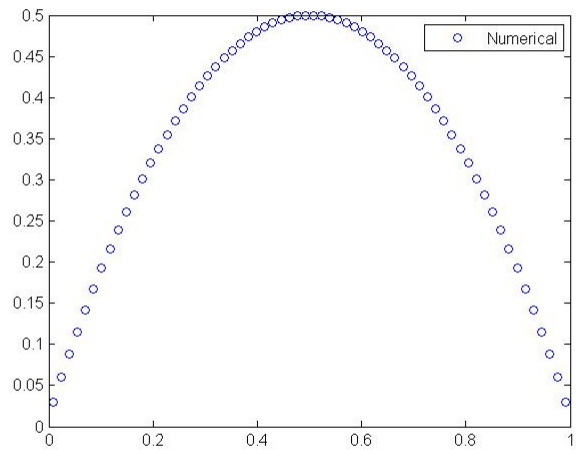


Figure 7: Showing the comparison between approximate FDM solutions and exact solutions for proposed example when $n = 128$

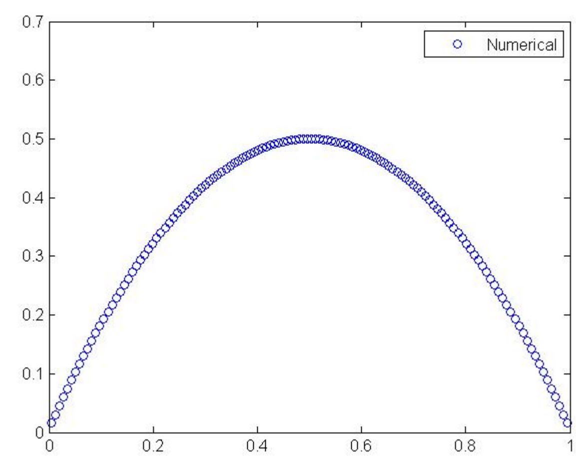


Figure 8: Showing the comparison between approximate FDM solutions and exact solutions for proposed example when $n = 256$

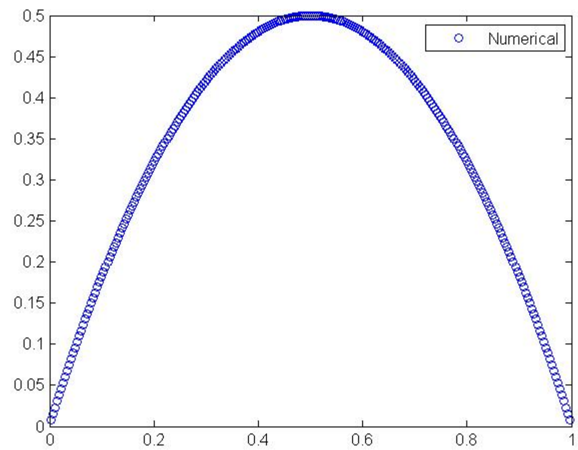


Figure 9: Showing the comparison between approximate FDM solutions and exact solutions for proposed example when $n = 512$

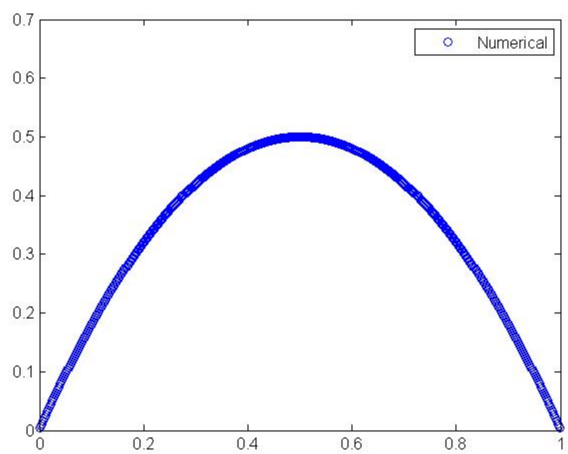
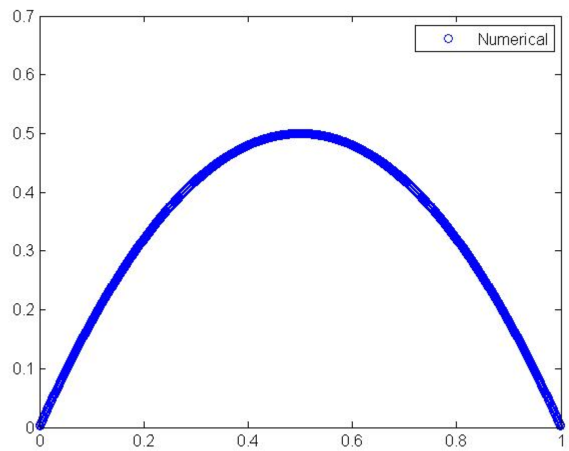


Figure 10: Showing the comparison between approximate FDM solutions and exact solutions for proposed example when $n = 1024$



we have calculated that elapsed time with $h = 1/1024$ was 3273.442931 seconds. On the other hand, elapsed time with $h = 1/64$ was 1.076592 seconds. We don't have any exact solutions for this proposed problems. But analyzing our figures, we can conclude that we have some problems in defining boundary conditions in our code. It is also encouraging for us that our $MATLAB$ code for our previous test problem has worked perfectly. Therefore, it will guide us to fix our code for proposed problem.

4. Conclusion

In this project work, It has been shown that matrix A is a sparse matrix which is usually true of finite difference methods for third-order boundary value problems. Since the matrix A has lots of zero entries, so in our $MATLAB$ code we have stored A as a *sparse* matrix. This means that the locations of nonzero entries, and the matrix entries at those locations, are stored; this saves space. Also there are "expert systems" in $MATLAB$ which recognize sparsity and then try to exploit it to speed up matrix/vector operations. In this project, we have used Gauss elimination method to solve our linear system.

The difference method used in this project work has local truncation error of order $O(h^2)$. To obtain a difference method with greater accuracy, we can proceed in a number of ways. Using fifth-order Taylor series for approximating truncation error term involving $O(h^4)$. However, this process has some in-built deficiency and the solution to the system requires many more calculations. Instead of attempting to obtain a difference method with a higher-order truncation error in this manner, it is generally more satisfactory to consider a reduction in step size. In addition, Richardson's extrapolation technique can be used effectively for this method because the error term is expressed in even powers of h with coefficients independent of h, provided $u(x)$ is sufficiently differentiable. We can conclude that FDM provides solutions only at the chosen grid points. We have obtained more accurate results for smaller step size using FDM.

References

[1] F. Bernis and L. A. Peletier, *Two problems from draining flows involving third-order ordinary differential equations*, SIAM J. Math. Anal., **27**(1996), 515-527.

[2] J. S. Guo and J. C. Tsai, *The structure of solutions for a third-order differential equation in boundary layer theory*, Japan J. Industrial Appl. Math., **22**(2005), 311-351.

[3] J.H. He, *Variational iteration method – a kind of nonlinear analytical technique: some examples*, Int. J. Nonlinear Mech., **34**(1999), 699-708.

[4] W.C. Troy, *Solutions of third-order differential equations relevant to draining and coating flows*, SIAM J. Math. Anal., **24**(1993), 155-171.

[5] E.O. Tuck and L.W. Schwartz, *A numerical and asymptotic study of some third-order ordinary differential equations relevant to draining and coating flows*, SIAM Review, **32**(1990), 453-469.

[6] Cheney and Kincaid.*Numerical Mathematics and Computing*, 7th Edition.