SOLVE BOUNDARY VALUE PROBLEM OF SHOOTING AND FINITE DIFFERENCE METHOD USING MATLAB

Sheikh Md. Rabiul Islam

Department of Electronics & Communication Engineering
Khulna University of Engineering &Technology, Khulna , Bangladesh.

Email: robi_kuet@yahoo.com; robi_ece@ece.kuet.ac.bd

Abstract: *In this paper, the order of convergence of finite difference methods& shooting method has been presented for the numerical solution of a two-point boundary value problem (BVP) with the second order differential equations (ODE's) and analyzed. Sufficient condition guaranteeing a unique solution of the corresponding boundary value problem is also given. Numerical results are tabulated for typical numerical examples and compared with the shooting technique employing the classical Euler and fourth-order Runge-Kutta method using MATLAB 7.6.0(R2008a).*

Keywords: *BVP, Shooting method, Finite difference method, MATLAB, Euler method, Runge-Kutta method.*

1. Introduction

These problems are called two-point boundary value problems and formally have the form [1]

$$\left.\begin{aligned} ODE: y''(t) &= s(x, y(x), y'(x)), x \in (a,b) \\ BCs: y(a) &= \alpha , y(b) = \beta \end{aligned}\right\}$$

$$\left.\begin{aligned} ODE: y''(t) &= s(x, y(x), y'(x)), x \in (a,b) \\ BCs: y(a) &= \alpha , y(b) = \beta \end{aligned}\right\} \quad \dots\dots (1)$$

where α and β are prescribed real values. The second order ODE may be linear or non-linear, depending on the function s. The linear version of the BVP (1) is obtained by choosing the function $s\left(x, y(x), y'(x)\right)$ to have a particular form, namely

$$\left.\begin{aligned} ODE: y''(x) &= p(x)y'(x) + q(x)y(x) + r(x), x \in [a,b] \\ BCs: y(a) &= \alpha , y(b) = \beta \end{aligned}\right\} \quad \dots\dots (2)$$

where the coefficients p(x), q(x) and r(x) are prescribed functions of time, or constants. The following theorem assures existence and uniqueness of the solution of the non-linear BVP (3).

Consider the Boundary Value Problem (BVP) of theODE:

$$\epsilon y''(x) + y'(x) = 1; y(0) = y(1) = 0 \quad \dots\dots (3)$$

Perform (1) applying the following methods with N = 10; 100; 1000; 10000 mesh points and damping coefficients $\epsilon = 1, 0.1, 10^{-4}, 10^{-6}$ using Finite difference method &Shooting method with Euler & Runge Kutta 4th order as forward integrator.

The exact solution is

$$y(x) = x - \frac{1 - \exp(-x\epsilon^{-1})}{1 - \exp(-\epsilon^{-1})} \quad \dots\dots (4)$$

The authors investigated an estimation in numerically the order of convergence in BVP after when ϵ is small

2. Numerical Method Analysis

A. The Euler Method

Explicit Euler's method [1] is the simplest case of a Taylor method, where only the first term of the increment function is used, with second and higher order terms neglected.

The method is as follows:

$$y^{n+1} = y^n + hs\left(x^n, y^n\right) \quad \dots\dots (5)$$

Where, $s\left(x^n, y^n\right)$is the source term .

The Euler's method is very simple to use but accuracy can get only first-order solution.

B. The Fourth Oder Runge-Kutta method

This is a popular higher order numerical method[1] . in particular ,it is a fourth order accurate method whose scheme is:

$$y^{n+1} = y^n + h\phi(x^n, y^n, h) \quad \dots\dots (6)$$

$$\phi(x^n, y^n, h) = \frac{1}{2(k_1 + 2k_2 + 2k_3 + k_4)}$$

$$k_1 = s(x^n, y^n)$$

$$k_2 = s\left(x^n + h, y^n + hk_1\right)$$

$$k_3 = s\left(x^n + \frac{1}{2}h, y^n + \frac{1}{2}hk_2\right)$$

$$k_4 = s\left(x^n + h, y^n + hk_3\right)$$

Date of submission : 15.03.2011 Date of acceptance : 07.02.2012

3. Shooting Method

The shooting method begins by associating to the original BVP (3) an IVP of the form[1]

$$ODE: y''(t) = s(t, y(t), y'(t)), t \in (a, b) \quad \ldots(7)$$
$$BCs: y(a) = \alpha, y'(b) = g$$

where g is a parameter that determines an initial guess for the *slope of the curve* $y(t)$. This associated IVP (7) is solved by the usual methods for IVPs, we have considered the two methods which is Euler and fourth-order Runge-Kutta method.

- For two chosen initial slopes $g1$, $g2$ compute two solutions to IVP (7), with corresponding boundary points $(g1, B(g1))$ and $(g2, B(g2))$.
- Fit a straight line through the points $P1 = (g1, B(g1))$ and $P2 = (g2, B(g2))$, namely[1]

$$B(g) = B(g_2) - \left\{\frac{B(g_2) - B(g_1)}{g_2 - g_1}\right\}(g - g_2)$$

A third value $g3$ is obtained by requiring $B(g_3) = \beta$, that is

$$g_3 = g_2 - \left\{\frac{g_2 - g_1}{B(g_2) - B(g_1)}\right\}(\beta - B(g_2))$$

$$g_{k+1} = g_k - \left\{\frac{g_k - g_{k-1}}{B(g_k) - B(g_{k-1})}\right\}(\beta - B(g_k)), k = 2, \ldots, K \quad \ldots(8)$$

- The process is stopped if

$$|B(g_k) - \beta| \le TOL \qquad (9)$$

where TOL is a tolerance, a preassigned small positive real number. For single precision calculations we can take $TOL = 10^{-6}$.

4. Numerically Survey of Linear Shooting Method

Looking at problem class (2), we break this down into two IVP also shown in

$$y_1''(x) = p(x)y_1' - q(x)y_1 + r(x), a \le x \le b,$$
$$y_1(a) = \alpha \quad y_1'(a) = 0 \qquad (10)$$
$$y_2''(x) = p(x)y_2' - q(x)y_2, \quad a \le x \le b,$$
$$y_2(a) = 0 \quad y_2'(a) = 1 \quad \ldots\ldots \quad (11)$$

Combining these results together to get the unique solution

$$y(x) = y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x) \quad \ldots\ldots(12)$$

Provided that $y_1(b) \ne 0$.

From the BVP of equation (3) can be written as

$$y''(x) = -\frac{1}{\epsilon}y'(x) - \frac{1}{\epsilon}$$

With boundary conditions

$$y(0) = 0$$
$$y(1) = 1 - \frac{1 - \exp(-\epsilon^{-1})}{1 - \exp(-\epsilon^{-1})} \quad \ldots\ldots \quad (13)$$

Breaking this boundary value problem into two IVP's

$$y_1'' = -\frac{1}{\epsilon}y_1' - \frac{1}{\epsilon}, \quad y_1(a) = 0 \quad y_1'(a) = 0 \quad \ldots(14)$$
$$y_2''(x) = -\frac{1}{\epsilon}y_2' - \frac{1}{\epsilon}, \quad y_2(a) = 0 \quad y_2'(a) = 1 \ldots \quad (15)$$

Discretising (14) let consider again

$$y_1 = z_1 \quad y_1' = z_2$$
$$z_1' = z_2 \quad z_1(a) = 0 \qquad \ldots\ldots \quad (16)$$
$$z_2' = -\frac{1}{\epsilon}z_2 - \frac{1}{\epsilon}, \quad z_2(a) = 0 \quad \ldots\ldots(17)$$

Using the Euler method we have the two difference equations [1];[2]

$$z_{1i+1} = z_{1i} + hz_{2i} \qquad \ldots\ldots \quad (18)$$
$$z_{2i+1} = z_{2i} + h\left(-\frac{1}{\epsilon}z_{2i} + \frac{1}{\epsilon}\right)\ldots\ldots \quad (19)$$

Similarly using the fourth order of Runge-Kutta method [1];[2] we have two difference equations

$$y^{n+1} = y^n + h\Phi(x^n, y^n, h)\ldots\ldots \quad (20)$$
$$\Phi(x^n, y^n, h) = \frac{1}{2}(k_1 + 2k_2 + 2k_3 + k_4) \qquad \ldots(21)$$
$$z_{1i+1} = z_{1i} + hz_{2i} \quad \ldots\ldots \quad (22)$$
$$k_1 = s(x^n, y^n) = -\frac{1}{\epsilon}z_{2i} + \frac{1}{\epsilon} \ldots. \quad (23)$$
$$k_2 = s(x^n + h, y^n + hk_1) = -\frac{1}{\epsilon}(z_{2i} + hk_1) + \frac{1}{\epsilon}$$
$$k_3 = s\left(x^n + \frac{1}{2}h, y^n + \frac{1}{2}hk_1\right) = -\frac{1}{\epsilon}\left(z_{2i} + \frac{1}{2}hk_1\right) + \frac{1}{\epsilon}$$
$$k_4 = s(x^n + h, y^n + hk_3) = -\frac{1}{\epsilon}(z_{2i} + hk_3) + \frac{1}{\epsilon}$$
$$z_{2i+1} = z_{2i} + h\frac{1}{2}(k_1 + 2k_2 + 2k_3 + k_4) \quad \ldots \quad (24)$$

Discretising (15) let consider

$$y_1 = w_1 \quad y_1' = w_2 \qquad \ldots\ldots. \quad (25)$$
$$w_1' = w_2 \quad w_1(a) = 0 \qquad \ldots\ldots. \quad (26)$$
$$w_2' = -\frac{1}{\epsilon}w_2 + \frac{1}{\epsilon}, \quad w_2(a) = 1 \quad \ldots \quad (27)$$

Using the Euler method we have the two difference equations

$$w_{1i+1} = w_{1i} + hw_{2i} \quad \ldots\ldots \quad (28)$$
$$w_{2i+1} = w_{2i} + h\left(-\frac{1}{\epsilon}w_{2i} + \frac{1}{\epsilon}\right)\ldots\ldots \quad (29)$$

Similarly using the fourth order of Runge-Kutta method we have two difference equations

$$y^{n+1} = y^n + h\Phi(x^n, y^n, h) \quad \ldots\ldots \quad (30)$$
$$\Phi(x^n, y^n, h) = \frac{1}{2}(k_1 + 2k_2 + 2k_3 + k_4) \quad \ldots(31)$$
$$w_{1i+1} = w_{1i} + hw_{2i}$$
$$k_1 = s(x^n, y^n) = -\frac{1}{\epsilon}w_{2i} + \frac{1}{\epsilon}$$
$$k_2 = s(x^n + h, y^n + hk_1) = -\frac{1}{\epsilon}(w_{2i} + hk_1) + \frac{1}{\epsilon}$$

$$k_3 = s\left(x^n + \frac{1}{2}h, y^n + \frac{1}{2}hk_1\right) = -\frac{1}{\epsilon}\left(w_{2i} + \frac{1}{2}hk_1\right) + \frac{1}{\epsilon}$$
$$k_4 = s(x^n + h, y^n + hk_3) = -\frac{1}{\epsilon}(w_{2i} + hk_3) + \frac{1}{\epsilon}$$
$$w_{2i+1} = w_{2i} + h\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \qquad (32)$$

Combing all these to get our solution using Euler method and fourth order Runge-Kutta method

$$y_i = z_{1i} + \frac{\beta - z_1(b)}{w_1(b)} w_{1i} \quad \dots\dots \ (33)$$

We divided the area into even spaced mesh points

$$x_0 = a, x_N = b, x_i = x_0 + ih \ ; h = \frac{b-a}{N} \ \dots (34)$$

We have $N = 10, 100, 1000, 10000 \text{ and } a = 0 \text{ and } b = 1$ and computation algorithm of the above is computationally complex and to solve it for Shooting method using Euler and fourth order of Runge-Kutta method to find the hit to target value of β with the some initial guess consider two problem as shown in equation (13) and (14). In BVP of equation we have also used the value of $\epsilon = 1, 0.1, 10^{-4}, 10^{-8}$ corresponding shown Table 1 for Shooting method using Euler and fourth order of Runge-Kutta method. We have also tried to find the order of convergence using the equation (2) for different of N for both Euler method and Fourth order Runge-Kutta method. We have observed in Table 1 when $\epsilon = 1$ the error has gradually decreased as well as the order of convergence has decreased within the mesh size increased the order of convergence 1st order for Euler method and the order of convergence is 4 for N=100 as well as the error rate is quite low in 4th Oder Runge-Kutta method. In Table 1 for $\epsilon = 0.1$, the order of convergence 1st order for Euler method and 4th Oder for 4th order of Runge-Kutta method and also error rate gradually decreases within the increase of mesh size N . In Table 1 when $\epsilon = 10^{-4}$ the Euler method has shown the error enlargement in size and not a number (NaN) of order of convergence within a increases the mesh size N for both the Euler and Fourth order Runge-Kutta method. In Table 1, for $\epsilon = 10^{-8}$, it has shown that the error rate to become greater or more in size for the fixed of the mesh size and order of convergence shown not a number(NaN) for both of Euler method and 4th order of Runge-Kutta method.

5. The Method of Finite Differences

Each finite difference operator can be derived from Taylor expansion. Once again looking at a linear second order difference equation

$$y'' = p(x)y' + q(x)y + r(x) \quad \dots\dots \ (35)$$

On $[a, b]$ subject to boundary conditions

$$y(a) = \alpha \ , y(b) = \beta \quad \dots\dots \ (36)$$

As with all the case we divide the area into even spaced mesh points

$$x_0 = a, x_N = b, x_i = x_0 + ih \ ; h = \frac{b-a}{N} \qquad (37)$$

For any function *y*(*x*), with $x \in [a, b]$, one can define point values $y_i = y(x_i)$. If *y*(*x*) is sufficiently smooth we

Table 1: Computation Results for Shooting Method using Euler Method and Fourth Order Runge-Kutta Method when $\epsilon = 1, 0.1, 10^{-4}, 10^{-8}$.

			$\epsilon = 1$		$\epsilon = 0.1$	$\epsilon = 10^{-4}$			$\epsilon = 10^{-8}$	
	Methods	Number of Mesh size(N)	Error	Oder of convergence(p)	Error	Oder of convergence(p)	Error	Oder of convergence(p)	Error	Oder of convergence(p)
Shooting Method	Euler	10	0.00645739		0.367851		0		0	
		100	0.000608022	1.02614	0.0191896	1.28261	0	NaN	0	NaN
		1000	6.04472e-005	1.00254	0.00184577	1.01689	0	NaN	0	NaN
		10000	6.04119e-006	1.00025	0.000183882	1.00164	0.367879	$-\infty$	0	NaN
	4th order of Runge-Kutta	10	1.09285e-007		0.00711489		0		0	
		100	1.01523e-011	4.032	3.32996e-007	4.32973	0	NaN	0	NaN
		1000	1.40166e-015	3.85992	3.0889e-011	4.03264	0	NaN	0	NaN
		10000	3.42365e-014	-1.38785	7.51536e-014	2.61385	0.00712056	$-\infty$	0	NaN

can also define approximations to the derivatives of $y(x^i)$ at any point x^i.

We now replace the derivatives $y'(x)$ and $y''(x)$ with the centered difference approximations from Taylor's theorem [1]

$$y'(x_i) = \frac{1}{2h}\left[y(x_{i+1}) - y(x_{i-1})\right] + O(h^2) = \frac{1}{2h}\left[y(x_{i+1}) - y(x_{i-1})\right] + O(h^2) \quad \ldots\ldots(38)$$

$$y''(x_i) = \frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] = \frac{1}{h^2}\left[y_{i+1} - 2y_i + y_{i-1}\right] \quad \ldots\ldots(39)$$

for $i = 1, \ldots\ldots, N-1$

We now have the equation

$$\frac{1}{h^2}\left[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\right] = p(x_i)\frac{1}{2h}\left[y(x_{i+1}) - y(x_{i-1})\right] + q(x_i)y(x_i) + r(x_i)$$

for $i = 1, \ldots\ldots, N-1$. …… (40)

Since the values of $p(x_i), q(x_i)$ *and* $r(x_i)$ are known it represents linear algebraic equation involving $y(x_{i-1}), y(x_i), y(x_{i-1})$.

Recall that

$y(a) = y_0 = \alpha, y(b) = y_{N+1} = \beta.$

Rearranging equation (20) we get the expression

$$-\left(1 + \frac{hp(x_i)}{2}\right)y_{i-1} - \left(2 + h^2 q(x_i)\right)y_i - \left(1 - \frac{hp(x_i)}{2}\right)y_{i+1} = h^2 r(x_i) \quad \ldots\ldots(41)$$

The values of $y_i, (i = 1, \ldots\ldots, N-1)$ can therefore be found by solving the traditional system $Ay = B$

$$A = \begin{bmatrix} 2+h^2q(x_1) & -1+\frac{hp(x_1)}{2} & 0 & \cdots & \cdots & 0 \\ -1-\frac{hp(x_2)}{2} & 2+h^2q(x_2) & -1+\frac{hp(x_2)}{2} & \ddots & & \vdots \\ 0 & -1-\frac{hp(x_3)}{2} & 2+h^2q(x_3) & -1+\frac{hp(x_3)}{2} & \ddots & \\ \vdots & 0 & \ddots & \ddots & & \\ & 0 & & & & 0 \\ & & -1-\frac{hp(x_{N-1})}{2} & 2+h^2q(x_{N-1}) & -1+\frac{hp(x_{N-1})}{2} & \\ 0 & \cdots & \cdots & 0 & -1-\frac{hp(x_N)}{2} & 2+h^2q(x_N) \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \vdots \\ y_{N-1} \\ y_N \end{bmatrix}, B = \begin{bmatrix} -h^2r(x_1) + \left(1 + \frac{hp(x_1)}{2}\right)\alpha \\ -h^2r(x_2) \\ -h^2r(x_3) \\ \vdots \\ \vdots \\ -h^2r(x_{N-1}) \\ -h^2r(x_N) + \left(1 + \frac{hp(x_N)}{2}\right)\beta \end{bmatrix}$$

6. Numerically Survey of Finite Difference Method

Looking at the BVP of equation (1) with the exact equation

$$y''(x) = -\frac{1}{\epsilon}y'(x) + \frac{1}{\epsilon}, y(0) = y(1) = 0; \quad \ldots.(42)$$

The difference equation is of the form

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} = p(x_i)\left[\frac{y_{i+1}-y_{i-1}}{2h}\right] + q(x_i) + r(x_i) = -\frac{1}{\epsilon}\left[\frac{y_{i+1}-y_{i-1}}{2h}\right] + \frac{1}{\epsilon} \quad \ldots(43)$$

In the matrix form to find diagonal linear system which is much more computationally complex in a paper sheet and try to solve the computation burden and also calculate the error and order of convergence. In Table 2(a) the error rate has increased with the exact solution if the value of ϵ is much more small as well as the order of convergence to increase one's possessions as in Table 2 for $\epsilon = 10^{-4}, 10^{-8}$, but in the order of convergence is very much small compare to $\epsilon = 1, 0.1$

Table 2: Computation Results for Finite Difference Method when $\epsilon = 1, 0.1, 10^{-4}, 10^{-8}$

		$\epsilon = 1$		$\epsilon = 0.1$		$\epsilon = 10^{-4}$		$\epsilon = 10^{-8}$	
	Number of Mesh size(N)	Error	Oder of convergence(p)	Error	Oder of convergence	Error	Oder of convergence	Error	Oder of convergence
Finite Difference Method	10	0.000100686		0.0345287		49.9048		500000	
	100	1.0068e-006	2.00003	0.000306674	2.0515	0.997347	1.6993	4999.99	2
	1000	1.0068e-008	2	3.06344e-006	2.00047	0.666712	0.174908	50.0056	1.99995
	10000	1.01647e-010	1.99585	3.0633e-008	2.00002	0.0345461	1.28554	1.03691	1.68328

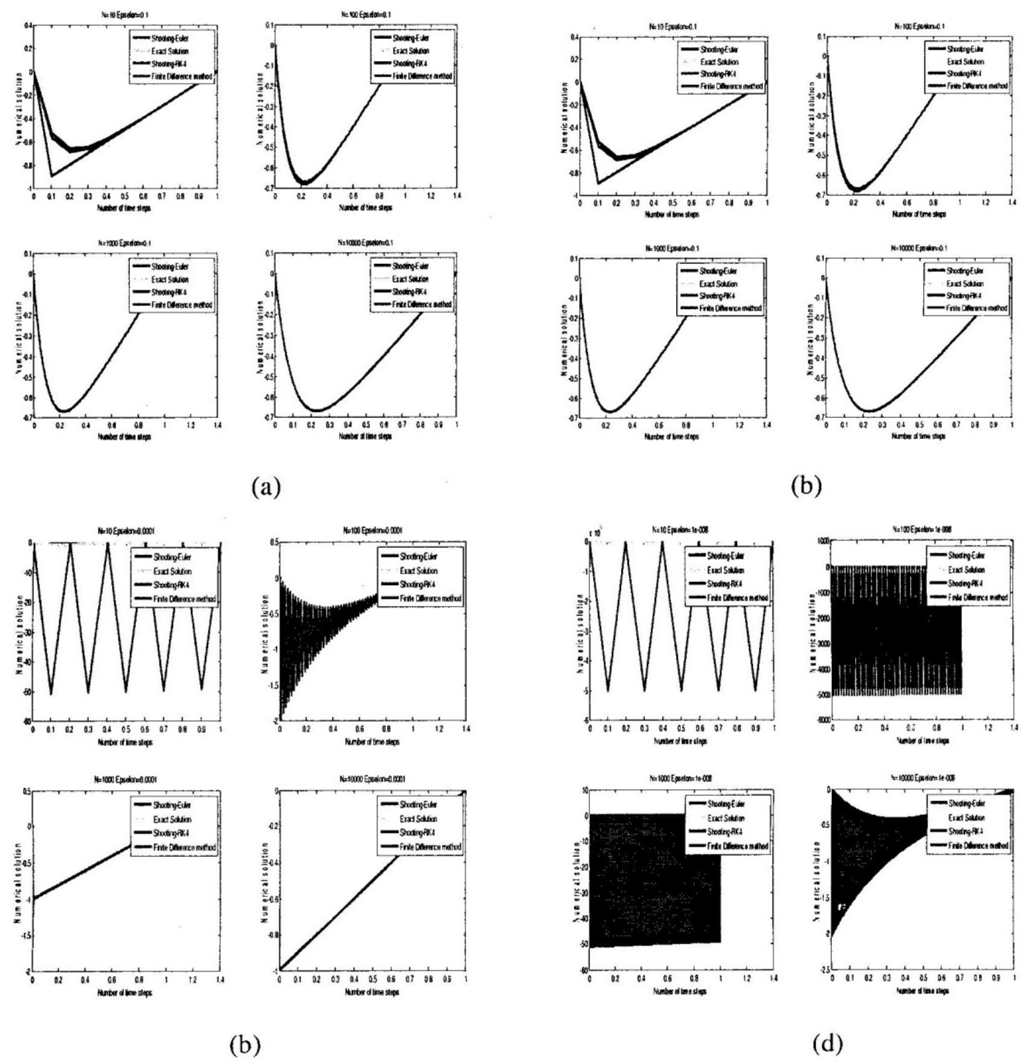

Fig.1. BVP for Finite difference method and Shooting method using Euler & 4th order Runge-Kutta method with exact solution when (a) $\epsilon = 1$ (b) $\epsilon = 0.1$ (c) $\epsilon = 10^{-4}$ (d) $\epsilon = 10^{8}$ and mesh size N=10, 100,1000,10000 with the boundary value is a=0 and b=1.

7. Measurements Analysis of BVP

For a fixed ϵ the BVP of (3) in the boundary region [0,1] has been solved. We observed in Fig.1 the Finite difference method and the exact solution is equal that is invisible with eyes, also observed the shooting method using Euler method are slightly far as well as using Runge-Kutta method from the exact solution are invisible to eyes because of the error rate are comparatively higher than Finite difference method for $\epsilon = 1$ with N=10 .We have investigated deeply insight in Fig. 1 (a) & (b) the value of ϵ consider much more small as1 and 0.1 with increase of mesh size N the error rate with exact solution as well as other numerical solution is quit low. In this situation numerical solution of shooting method and have not been visible after that small value of ϵ the numerical solution incapable of being seen and closely .In Fig.1(c) & (d) we see that we have seen the waves running in step with just a small difference in amplitude and phase resulting from small value of ϵ with the increases of N and the signal shown as instability condition for numerical solution. In Finite difference method, consider $\epsilon = 1$ in Table II the error has decreases with the increases of N where the order of accuracy for finite difference method is the 2nd order for $\epsilon = 1$& 0.1. We have also seen in the Table.II has created much more and more error for ϵ small. The error rate is high the stability of finite difference method to create an oscillation becomes an unstable as shown in Fig.1(d) an Figure.4.It can be seen from the numerical results presented in the previous section that the shooting method produces good approximation solution to BVP. It may be observed that the initial conditions assigned to new problems (derived from the original problem) are obtained easily from the solution of reduced problem.

8. Conclusions

We introduced an order of convergence of shooting and finite difference method for a general BVP. We have verify in Table I-II the theatrical analysis of the design and rate of convergence is close to four for Runge-Kutta method and one for Euler method and also close to two for finite difference method. It's shown minimum error for all methods for damping coefficient $\epsilon = 1$&0.1 and higher error for ϵ taken smaller in this manner numerically diffused for both methods.

References

[1] Toro E.F., Riemann, "Solvers and Numerical methods for fluid dynamics". Second edition, Springer 1999.

[2] A. A Slama,A.A.Mansour "Fourth-order Finite - differnce method for third-order boundary-value problem"Numerical Heat Transfer,Part B,47:383-401,2005.

[3] Lambert J D. Numerical methods for ordinary differential systems. The initial value problem.John Wiley and Sons, 1997.

[4] Shampine L F, Gladwell I and Thompson S. Solving ODEs with MATLAB. Cambridge University Press, 2003.

[5] Randall J. LeVeque "Finite Difference Methods for Differential Equations" AMath 585, Winter Quarter 2006 University of Washington Version of January, 2006.

[6] Zhilin Li, "Finite Difference Methods Basics" Center for Research in Scienti¯c Computation & Department of Mathematics North Carolina State University Raleigh, NC 27695.

[7] Bosho S. Jovanovic "Finite differenc schemes for boundary value problems with gereralized solution" NOVI SAD J. MATH VOL.30,No.3,2000,45-58

[8] Riaz A Usmani "Finite difffernce methods for a certain two point boundary value problem" Indian J. pure appl. Math., 14(3): 398-411,March 1983.

[9] Raymond Hollsopple, Ram Venkataraman, David Doman "A modified Sipmle Shooting Method for Solving Two-Point Boundary-Value Problems"

Sheikh Md. Rabiul Islam received the B.Sc.in Engg. (ECE) from Khulna University, Khulna, Bangladesh in December 2003, and M.Sc. in Telecommunication Engineering from the University of Trento, Italy, in October 2009 an also going an M.Sc in Engineering (EEE) under the Khulna University of Engineering & Technology, Khulna_9203, Bangladesh. He joined as a Lecturer in the department of electronics and communication engineering of Khulna University of Engineering & Technology, Khulna, in 2004, where he is currently an Assistant Professor in the same department in the effect of 2008. He has published two Journal and four conferences in national and International (Specially USA) level. Since 2004 to 2006 he has acted as a visiting lecturer of ECE and CSE discipline in Khulna University, Bangladesh. His research interests include antennas and propagation, wireless communications & signal processing, and biomedical engineering. Mr. Islam is an Associate Member of Institute of Engineers Bangladesh (IEB) and Life member of Bangladesh Electronic Society (BES).