International Journal of Physics, Chemistry and Mathematics
Vol. 2, No. 1, March 2014, PP: 1 – 16
Available online at http://ijpcm.com/

Research article

Numerical Integration Method for Solving Two Point Boundary Value Problems by Simpson's 1/3 rule

***TsegayTesfay[1], GetachewAdamu (Dr.)[2]**

[1]AssosaUniversity,P.O.Box 18, Assosa, Ethiopia , [2]Bahir-Dar University,P.O.Box 26, Bahir-Dar, Ethiopia

Corresponding Address:TsegayTesfay

E-mail: tsgytsfy@yahoo.com

Abstract

In this paper, a numerical integration method is presented for solving second order linear differential equation with two point boundary value problem. The original second order differential equation is replaced by an approximate first order differential equation with a small deviating argument. Then, the Simpson's one third rule is used to approximate the definite integral and to obtain the three term recurrence relationship. The method is iterative on the deviating argument. To demonstrate the applicability of the method, number of example is solved. And also it tries to outline the numerical integration method for the computation and comparing the approximate solution with the exact solution.

Keywords: Approximate solution; Boundary value; Condition; Numerical integration; Recurrence relation

---

Introduction

Historically, ordinary differential equations have originated in chemistry, physics and engineering. They still do, but nowadays they also originate in Medicine, Biology, Anthropology, and the like. Numerical integration used for solving two point boundary value problems which has been intuitively developed over many years. Its popularity is due to the applicability of boundary value problems in many engineering problems. Two point boundary value problems, which are ordinary differential equation with boundary conditions, occur in many Engineering and computational fields.

Solving such type of boundary value problems analytically is possible only in very rare cases. Many researchers worked for the numerical solutions of higher order boundary value problems. Some numerical methods such as finite difference method, shooting method, spline methods and others have been developed for solving such boundary value problems.

This report has two chapters; the first chapter includes the basic ideas and definitions which take part in understanding the second chapter. The second chapter contains; the derivations of the second order ordinary differential equation to first order, applying Simpson's one third rule for solving the first order ordinary differential equation, MATLAB programs, examples, and contains tables and graphs which compare the computational results to the analytical solution.

Numerical Integration Method for solving two point boundary Value problem

Let us consider the following linear two point boundary value problem

$$y'' + a(x)y' + b(x)y = f(x), \qquad a \le x \le b$$

$$y(a) = \alpha\,, \quad y(b) = \beta \qquad \ldots\ldots\ldots..(12)$$

Whereα andβ are given constants; $a(x)\,, b(x)$ and $f(x)$ are assumed to be sufficiently continuously differentiable functions in [a,b].

Let δ be a small positive argument ($0 < \delta \ll 1$) By the Taylor series expansion in the neighborhood of the point x we have

$$y(x-\delta) = y(x) - \delta y' + \tfrac{1}{2}\delta^2 y'' - \tfrac{1}{3!}\delta^3 y^3 + \cdots \ldots\ldots\ldots\ldots (13)$$

Consequently (12) is replaced by the following first order differential equation with the small deviating argument δ,

$$y(x-\delta) \cong y(x) - \delta y' + \tfrac{1}{2}\delta^2 y'' \qquad \text{(a truncated Taylor series)}$$

$$y(x-\delta) + \delta y'(x) - y(x) \cong \frac{1}{2}\delta^2 y''$$

$$\frac{2}{\delta^2}[y(x-\delta)+\delta y'(x)-y(x)] \cong y'' \qquad (14)$$

Substituting (14) in (12)

$$\frac{2}{\delta^2}[y(x-\delta)+\delta y'(x)-y(x)] + a(x)y'(x)+b(x)y(x) \cong f(x)$$

$$2y(x-\delta)+2\delta y'(x)-2y(x)+\delta^2 a(x)y'(x)+\delta^2 b(x)y(x) \cong \delta^2 f(x)$$

$$2y(x-\delta)+[2\delta+\delta^2 a(x)]y'(x)-[2+\delta^2 b(x)]y(x) \cong \delta^2 f(x)$$

$$[2\delta+\delta^2 a(x)]y'(x) \cong \delta^2 f(x)-[\delta^2 b(x)-2]y(x)-2y(x-\delta)$$

$$y'(x)=\frac{\delta^2 f(x)}{[2\delta+\delta^2 a(x)]}+\frac{[2-\delta^2 b(x)]}{[2\delta+\delta^2 a(x)]}y(x)-\frac{2}{[2\delta+\delta^2 a(x)]}y(x-\delta) \qquad (15)$$

We can rewrite equation (15) as

$$y' = p(x)y(x-\delta)+q(x)y(x)+r(x) \text{ for } a \le x \le b \qquad (16\text{ a})$$

Where

$$\left.\begin{aligned} p(x) &= \left[\frac{-2}{2\delta+\delta^2 a(x)}\right] \\ q(x) &= \left[\frac{2-\delta^2 b(x)}{2\delta+\delta^2 a(x)}\right] \\ r(x) &= \left[\frac{\delta^2 f(x)}{2\delta+\delta^2 a(x)}\right] \end{aligned}\right\} \qquad (16\text{ b})$$

Now let us divided the interval [a,b] in to N sub intervals with mesh size h, which is $h=\frac{b-a}{N}$, and the given interval must be divided in to even number of equal sub intervals, since we use two strips at a time.

Integrating equation (16 a) in $[x_i, x_{i+2}]$, for i=0, 1, 2, 3,…,N-2 we get

$$\int_{x_i}^{x_{i+2}} y'\,dx = \int_{x_i}^{x_{i+2}} \big(p(x)y(x-\delta)+q(x)y(x)+r(x)\big)\,dx \qquad (17)$$

By making use of the Simpson's 1/3 rule for evaluating the right side of equation (17) we obtain

$$y(x_{i+2})-y(x_i)=\int_{x_i}^{x_{i+2}} \big(p(x)y(x-\delta)+q(x)y(x)+r(x)\big)\,dx$$

$$= \frac{1}{3}h[p(x_i)y(x_i-\delta)+q(x_i)y(x_i)+r(x_i)] +$$

$$\frac{4}{3}h[p(x_{i+1})y(x_{i+1}-\delta)+q(x_{i+1})y(x_{i+1})+r(x_{i+1})] +$$

$$\frac{1}{3}h[p(x_{i+2})y(x_{i+2}-\delta)+q(x_{i+2})y(x_{i+2})+r(x_{i+2})]. \qquad \ldots\ldots\ldots\ldots(18)$$

Rearranging Equation (18) we can have the following

$$y(x_{i+2})-y(x_i) = \frac{1}{3}h[r(x_i)+4r(x_{i+1})+r(x_{i+2})] +$$

$$\frac{1}{3}h[q(x_i)y(x_i)+4q(x_{i+1})y(x_{i+1})+q(x_{i+2})y(x_{i+2})] +$$

$$\frac{1}{3}h[p(x_i)y(x_i-\delta)+4p(x_{i+1})y(x_{i+1}-\delta)+p(x_{i+2})y(x_{i+2}-\delta)]$$

$$\ldots..(19)$$

Where i=0,1,2,3,…,N-2 (We have N-1 number of equations)

From the Taylor series we have

$$y(x-\delta) \cong y(x)-\delta y'(x)$$

$$y(x_i-\delta) \cong y(x_i)-\delta y'(x_i) \qquad \ldots\ldots\ldots\ldots(20)$$

$$y(x_{i+1}-\delta) \cong y(x_{i+1})-\delta y'(x_{i+1}) \qquad \ldots\ldots\ldots\ldots..(21)$$

$$y(x_{i+2}-\delta) \cong y(x_{i+2})-\delta y'(x_{i+2}) \qquad \ldots\ldots\ldots\ldots..(22)$$

But from the idea of differentiation we can have the following

$$y'(x_i) \cong \frac{y(x_{i+1})-y(x_i)}{h} \qquad \ldots..\ldots\ldots(23)$$

$$y'(x_{i+1}) \cong \frac{y(x_{i+1})-y(x_i)}{h} \qquad \ldots\ldots\ldots(24)$$

$$y'(x_{i+2}) \cong \frac{y(x_{i+2})-y(x_{i+1})}{h} \qquad \ldots\ldots\ldots(25)$$

By substituting (23),(24) and (25) in (20),(21) and (22)Respectively we get

$$\left.\begin{array}{l} y(x_i-\delta) \cong y(x_i)-\delta\left(\frac{y(x_{i+1})-y(x_i)}{h}\right) \\ y(x_{i+1}-\delta) \cong y(x_{i+1})-\delta\left(\frac{y(x_{i+1})-y(x_i)}{h}\right) \\ \end{array}\right\} \qquad \ldots\ldots\ldots(26)$$

$$y(x_{i+2}-\delta) \cong y(x_{i+2}) - \delta\left(\frac{y(x_{i+2})-y(x_{i+1})}{h}\right)$$

Substituting (26) in (19) we have the following

$$y(x_{i+2}) - y(x_i) = \frac{1}{3}h[r(x_i) + 4r(x_{i+1}) + r(x_{i+2})] +$$

$$\tfrac{1}{3}h[q(x_i)y(x_i) + 4q(x_{i+1})y(x_{i+1}) + q(x_{i+2})y(x_{i+2})] +$$

$$\frac{1}{3}h\left[p(x_i)\left(y(x_i) - \delta\left(\frac{y(x_{i+1})-y(x_i)}{h}\right)\right) + 4p(x_{i+1})\left(y(x_{i+1}) - \delta\left(\frac{y(x_{i+1})-y(x_i)}{h}\right)\right) + p(x_{i+2})\left(y(x_{i+2}) - \delta\left(\frac{y(x_{i+2})-y(x_{i+1})}{h}\right)\right)\right]$$

$$y(x_{i+2}) - y(x_i) = \frac{1}{3}h[r(x_i) + 4r(x_{i+1}) + r(x_{i+2})] +$$

$$\tfrac{1}{3}h[q(x_i)y(x_i) + 4q(x_{i+1})y(x_{i+1}) + q(x_{i+2})y(x_{i+2})] +$$

$$\frac{h}{3}\left(p(x_i)\left(y(x_i)\left(1+\frac{\delta}{h}\right) - \frac{\delta}{h}y(x_{i+1})\right)\right) +$$

$$\frac{h}{3}4p(x_{i+1})\left[y(x_{i+1}) - \frac{\delta y(x_{i+1})}{h} + \frac{\delta y(x_i)}{h}\right]$$

$$+\tfrac{h}{3}p(x_{i+2})\left(\left(1-\tfrac{\delta}{h}\right)y(x_{i+2}) + \tfrac{\delta}{h}y(x_{i+1})\right) \qquad \ldots\ldots\ldots\ldots(27)$$

Rearranging (27) we have

$$-[r(x_i) + 4r(x_{i+1}) + r(x_{i+2})] = \left[q(x_i) + \frac{3}{h} + \left(1+\frac{\delta}{h}\right)p(x_i) + 4\frac{\delta}{h}p(x_{i+1})\right]y(x_i) +$$

$$\left[4q(x_{i+1}) - \frac{\delta}{h}p(x_i) + 4\left(1-\frac{\delta}{h}\right)p(x_{i+1}) + \frac{\delta}{h}p(x_{i+2})\right]y(x_{i+1})$$

$$+\left[\frac{-3}{h} + q(x_{i+2}) + \left(1-\frac{\delta}{h}\right)p(x_{i+2})\right]y(x_{i+2})$$

$$\ldots\ldots\ldots\ldots(28)$$

Let

International Journal of Physics, Chemistry and Mathematics
Vol. 2, No. 1, March 2014, PP: 1 – 16
Available online at http://ijpcm.com/

$$H_i = -[r(x_i) + 4r(x_{i+1}) + r(x_{i+2})]$$

$$E_i = \left[q(x_i) + \frac{3}{h} + \left(1 + \frac{\delta}{h}\right)p(x_i) + 4\frac{\delta}{h}p(x_{i+1})\right]$$

$$F_i = -\left[4q(x_{i+1}) - \frac{\delta}{h}p(x_i) + 4\left(1 - \frac{\delta}{h}\right)p(x_{i+1}) + \frac{\delta}{h}p(x_{i+2})\right]$$

$$G_i = \left[\frac{-3}{h} + q(x_{i+2}) + \left(1 - \frac{\delta}{h}\right)p(x_{i+2})\right]$$

So, (28) becomes

$$H_i = E_i y(x_i) - F_i y(x_{i+1}) + G_i y(x_{i+2}) \text{for i=0,1,2,…,N-2} \qquad \text{………… (29)}$$

The solution of the tri-diagonal system (29) can easily be obtained by using the “Thomas Algorithm”. In this algorithm we set a recurrence relation of the form

$$y_i = \omega_i y_{i+1} + \tau_i \qquad \text{………… (30)}$$

Where w_i and t_i are to be determined.

From (30)

$$y_{i+1} = \omega_{i+1} y_{i+2} + \tau_{i+1} \qquad \text{………… (31)}$$

$$H_i = E_i y_i - F_i y_{i+1} + G_i y_{i+2}$$

$$H_i = E_i(\omega_i y_{i+1} + \tau_i) - F_i y_{i+1} + G_i y_{i+2} \quad \text{Substituting (30) in (29)}$$

$$H_i = E_i\omega_i y_{i+1} + E_i\tau_i - F_i y_{i+1} + G_i y_{i+2}$$

$$H_i = E_i\omega_i y_{i+1} + E_i\tau_i - F_i y_{i+1} + G_i y_{i+2}$$

$$y_{i+1}(F_i - E_i\omega_i) = E_i\tau_i - H_i + G_i y_{i+2}$$

$$y_{i+1} = \left(\frac{G_i}{F_i - E_i\omega_i}\right) y_{i+2} + \left(\frac{E_i\tau_i - H_i}{F_i - E_i\omega_i}\right) \qquad \text{……… (32)}$$

Combining (31) and (32) we get

$$\left.\begin{aligned} \omega_{i+1} &= \frac{G_i}{F_i - E_i\omega_i} \\ \tau_{i+1} &= \frac{E_i\tau_i - H_i}{F_i - E_i\omega_i} \quad \text{for i= 1,2,…,n-1} \end{aligned}\right\} \qquad \text{……….(33)}$$

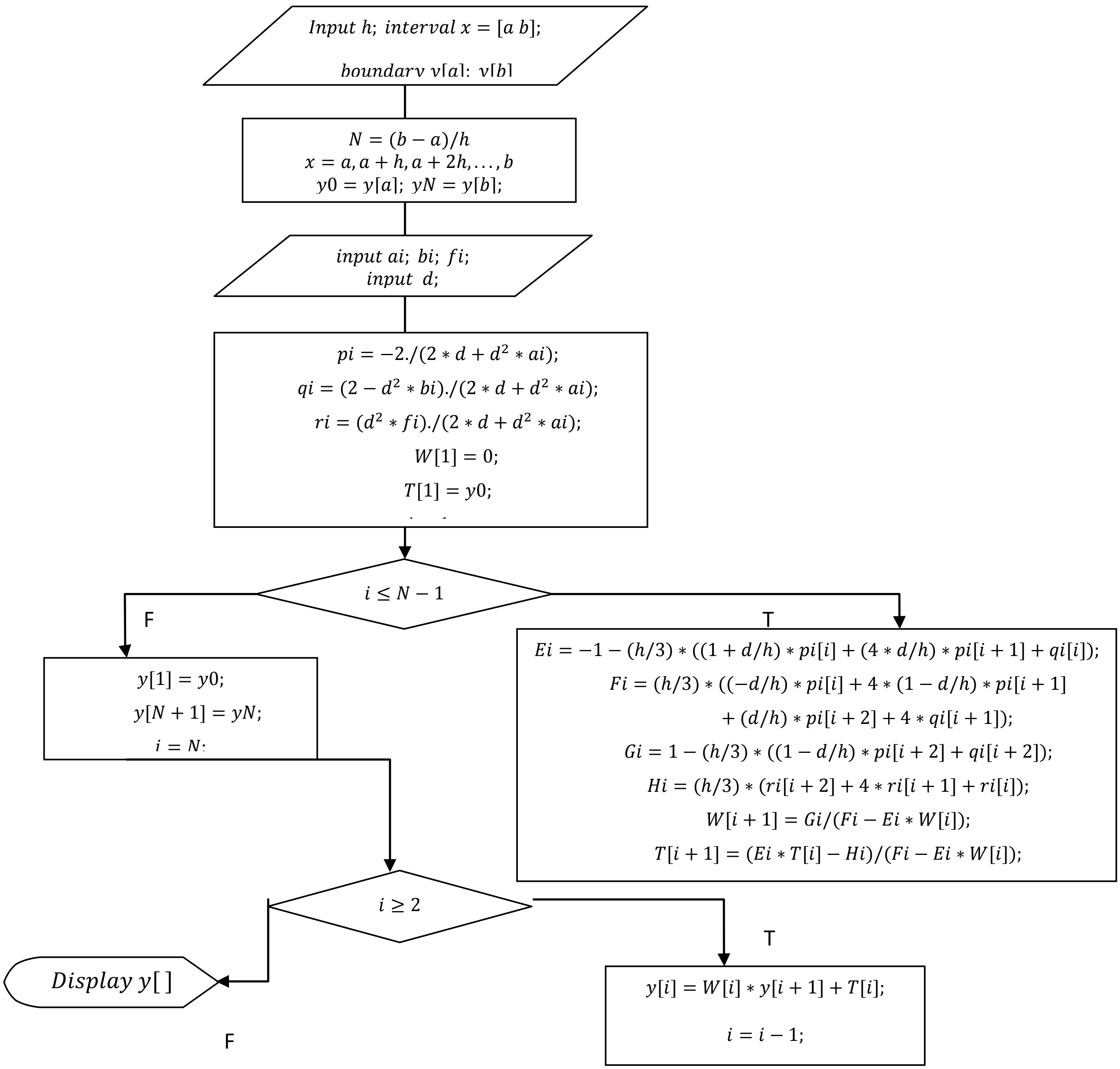
From the boundary conditions

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

$$\alpha = \omega_0 y_1 + \tau_0$$

If we choose ω_0=0, then $\alpha = \tau_0$ with these initial conditions we can get the values for $\omega_i \& \tau_i$ for i=0,1,2,3,…,N-2 from (33) in the forward process, then we can obtain y_i in the backward process of (30) using the boundary condition of (12).

Flow chart of case one



International Journal of Physics, Chemistry and Mathematics
Vol. 2, No. 1, March 2014, PP: 1 – 16
Available online at http://ijpcm.com/

Example 1: Consider the following non-homogenous linear differential equation of second order from (Rosłoniec, 2008)p. 219, Eg. 7.6] $y''-3y=2x$ and the boundary conditions y(0)=0,y(1)=1 where the exact solution is $y(x)=\frac{5\sinh(\sqrt{3}x)}{3\sinh(\sqrt{3})}-\frac{2x}{3}$

Solution: We have $a(x)=0$, b(x)=-3 and f(x)=$2x$ andfrom equation (16 b) we have

$$p(x)=\left[\frac{-1}{\delta}\right]\ ,\quad q(x)=\left[\frac{-3\delta^2-2}{2\delta}\right]\ ,\quad r(x)=x\delta$$

From equation (29) we have

$$H_i=E_iy(x_i)-F_iy(x_{i+1})+G_iy(x_{i+2}) \text{for i=0,1,2,…,N-2} \quad \text{where}$$

$$H_i=-[r(x_i)+4r(x_{i+1})+r(x_{i+2})]$$

$$E_i=\left[q(x_i)+\frac{3}{h}+\left(1+\frac{\delta}{h}\right)p(x_i)+4\frac{\delta}{h}p(x_{i+1})\right]$$

$$F_i=-\left[4q(x_{i+1})-\frac{\delta}{h}p(x_i)+4\left(1-\frac{\delta}{h}\right)p(x_{i+1})+\frac{\delta}{h}p(x_{i+2})\right]$$

$$G_i=\left[\frac{-3}{h}+q(x_{i+2})+\left(1-\frac{\delta}{h}\right)p(x_{i+2})\right]$$

For i=0

$$H_0=E_0y(x_0)-F_0y(x_1)+G_0y(x_2)$$

For i=1

$$H_1=E_1y(x_1)-F_1y(x_2)+G_1y(x_3)$$

For i=n-2

$$H_{n-2}=E_{n-2}y(x_{n-2})-F_{n-2}y(x_{n-1})+G_0y(x_n)$$

This results a tridiagonal matrix of $(n-1)\times(n-1)$ and it can be solved by Thomas algorithm manually as follows for h=0.1, n=10 and $\delta=0.05$

$$0.03=60.075y_0+120.3y_1+60.075y_2$$

$$0.06=60.075y_1+120.3y_2+60.075y_3$$

$$0.09 = 60.075y_2 + 120.3y_3 + 60.075y_4$$

$$0.12 = 60.075y_3 + 120.3y_4 + 60.075y_5$$

$$0.15 = 60.075y_4 + 120.3y_5 + 60.075y_6$$

$$0.18 = 60.075y_5 + 120.3y_6 + 60.075y_7$$

$$0.21 = 60.075y_6 + 120.3y_7 + 60.075y_8$$

$$0.24 = 60.075y_7 + 120.3y_8 + 60.075y_9$$

$$0.27 = 60.075y_8 + 120.3y_9 + 60.075y_{10}$$

In matrix form we have

$$\begin{bmatrix} 120.3 & 60.07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 60.07 & 120.3 & 60.07 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 60.07 & 120.3 & 60.07 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 60.07 & 120.3 & 60.07 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 60.07 & 120.3 & 60.07 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60.07 & 120.3 & 60.07 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 60.07 & 120.3 & 60.07 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 60.07 & 120.3 & 60.07 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 60.07 & 120.3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{bmatrix} = \begin{bmatrix} 0.03 \\ 0.06 \\ 0.09 \\ 0.12 \\ 0.15 \\ 0.18 \\ 0.21 \\ 0.24 \\ -59.805 \end{bmatrix}$$

Table 1: Computational results of Example 1 (a)

x(i)	Approx(y(i))
0.10	0.0511128
0.20	0.1048855
0.30	0.1640384
0.40	0.2314130
0.50	0.3100365
0.60	0.4031904
0.70	0.5144840
0.80	0.6479367
0.90	0.8080679

1.00	1.0000000

By using the matlab program we have the following results for different values of δ

Table 2: Computational results of Example 1 (b)

$h = 0.05$ and $\delta = 0.03$

x_i	Approx y(x_i)	Exact y(x_i)	Error err(x_i)
0.0000	0.0000000	0.0000000	0.0000000
0.1000	0.0438305	0.0393141	0.0045164
0.2000	0.0906529	0.0818154	0.0088375
0.3000	0.1435400	0.1307869	0.0127531
0.4000	0.2057287	0.1897062	0.0160225
0.5000	0.2807081	0.2623502	0.0183579
0.6000	0.3723133	0.3529084	0.0194049
0.7000	0.4848297	0.4661092	0.0187205
0.8000	0.6231089	0.6073619	0.0157470
0.9000	0.7927000	0.7829195	0.0097805
1.0000	1.0000000	1.0000667	0.0000667

$h = 0.05$ and $\delta = 0.035$

x_i	Approx y(x_i)	Exact y(x_i)	Error err(x_i)
0.0000	0.0000000	0.0000000	0.0000000
0.1000	0.0370990	0.0393141	0.0022151
0.2000	0.0774773	0.0818154	0.0043381
0.3000	0.1245180	0.1307869	0.0062689
0.4000	0.1818147	0.1897062	0.0078916
0.5000	0.2532849	0.2623502	0.0090653

0.6000	0.3432944	0.3529084	0.0096140
0.7000	0.4567947	0.4661092	0.0093145
0.8000	0.5994796	0.6073619	0.0078823
0.9000	0.7779655	0.7829195	0.0049540
1.0000	1.0000000	1.0000667	0.0000667

$h = 0.05$ and δ =0.0335

x_i	Approx y(x_i)	Exact y(x_i)	Error err(x_i)
0.1000	0.0390590	0.0393141	0.0002552
0.2000	0.0813156	0.0818154	0.0004998
0.3000	0.1300642	0.1307869	0.0007227
0.4000	0.1887956	0.1897062	0.0009106
0.5000	0.2613024	0.2623502	0.0010478
0.6000	0.3517939	0.3529084	0.0011145
0.7000	0.4650233	0.4661092	0.0010859
0.8000	0.6064315	0.6073619	0.0009304
0.9000	0.7823118	0.7829195	0.0006076
1.0000	1.0000000	1.0000667	0.0000667

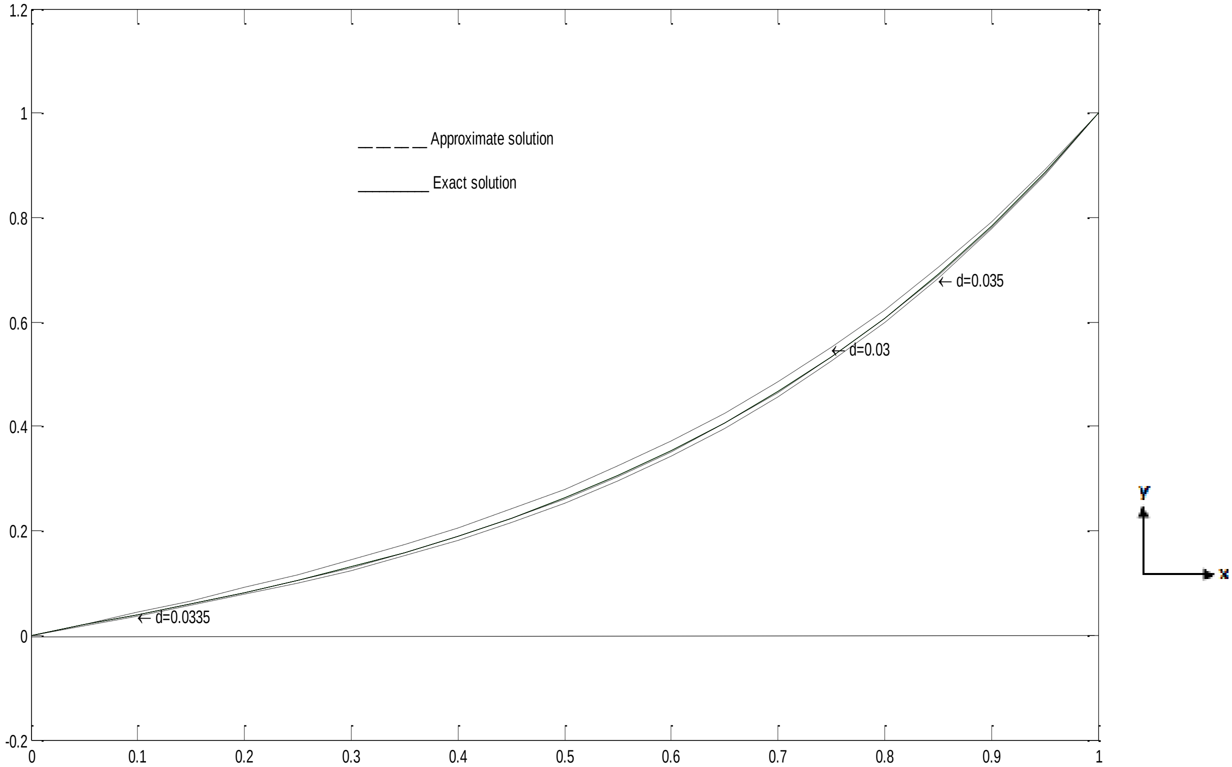


Figure 1: Computational results of Example 1 in graph

Example 3: consider the following homogenous linear differential equation of second order from [(Faires, 2005)p.665, Eq. 4c]

$y'' + \frac{4}{x}y' + \frac{2}{x^2}y = 2\frac{lnx}{x^2}$, $1 \leq x \leq 2$, $y(1) = 0.5, y(2) = 0.6931.$ Where the actual solution is

$$y = \frac{4}{x} + \frac{2}{x^2} - \frac{2}{x^2} + lnx - \frac{3}{2}$$

Solution

From equation (16 b) we have

$p(x) = \left[\frac{-2}{2\delta + \delta^2 a(x)}\right]$, $q(x) = \left[\frac{\delta^2 b(x) - 2}{2\delta + \delta^2 a(x)}\right]$,$r(x) = \left[\frac{\delta^2 f(x)}{2\delta + \delta^2 a(x)}\right]$ where

$a(x) = \frac{4}{x}$, b(x)=$\frac{2}{x^2}$ and f(x)=2$\frac{lnx}{x^2}$

From equation (29) we have

$H_i = E_i y(x_i) - F_i y(x_{i+1}) + G_i y(x_{i+2})$for i=0,1,2,…,N-2 where

$$H_i = -[r(x_i) + 4r(x_{i+1}) + r(x_{i+2})]$$

$$E_i = \left[q(x_i) + \frac{3}{h} + \left(1 + \frac{\delta}{h}\right) p(x_i) + 4\frac{\delta}{h} p(x_{i+1})\right]$$

$$F_i = \left[4q(x_{i+1}) - \frac{\delta}{h} p(x_i) + 4\left(1 - \frac{\delta}{h}\right) p(x_{i+1}) + \frac{\delta}{h} p(x_{i+2})\right]$$

$$G_i = \left[\frac{-3}{h} + q(x_{i+2}) + \left(1 - \frac{\delta}{h}\right) p(x_{i+2})\right]$$

For i=0

$$H_0 = E_0 y(x_0) - F_0 y(x_1) + G_0 y(x_2)$$

For i=1

$$H_1 = E_1 y(x_1) - F_1 y(x_2) + G_1 y(x_3)$$

For i=n-2

$$H_{n-2} = E_{n-2} y(x_{n-2}) - F_{n-2} y(x_{n-1}) + G_0 y(x_n)$$

This results a tridiagonal matrix of $(n-1) \times (n-1)$ and it can be solved by the matlab program,and the results are tabulated below.

Table 2: Computational results of Example 3

h=0.05 and $\delta = 0.03$

x_i	Approx y(x_i)	Exact y(x_i)	Error err(x_i)
1.0000	0.5000000	0.5000000	0.0000000
1.1000	0.5718741	0.5787813	0.0069071
1.2000	0.6174937	0.6267660	0.0092723
1.3000	0.6465034	0.6558554	0.0093520
1.4000	0.6648694	0.6732069	0.0083375
1.5000	0.6763785	0.6832429	0.0068643
1.6000	0.6834837	0.6887536	0.0052700
1.7000	0.6877989	0.6915279	0.0037290
1.8000	0.6903988	0.6927249	0.0023262
1.9000	0.6920041	0.6931004	0.0010963
2.0000	0.6931000	0.6931472	0.0000472

International Journal of Physics, Chemistry and Mathematics
Vol. 2, No. 1, March 2014, PP: 1 – 16
Available online at http://ijpcm.com/

h=0.05 and $\delta = 0.035$

x_i	Approx y(x_i)	Exact y(x_i)	Error err(x_i)
1.0000	0.5000000	0.5000000	0.0000000
1.1000	0.5826691	0.5787813	0.0038878
1.2000	0.6318664	0.6267660	0.0051004
1.3000	0.6608907	0.6558554	0.0050353
1.4000	0.6776049	0.6732069	0.0043979
1.5000	0.6867904	0.6832429	0.0035475
1.6000	0.6914186	0.6887536	0.0026650
1.7000	0.6933657	0.6915279	0.0018378
1.8000	0.6938291	0.6927249	0.0011041
1.9000	0.6935766	0.6931004	0.0004762
2.0000	0.6931000	0.6931472	0.0000472

h=0.05 and δ =0.0356

x_i	Approx y(x_i)	Exact y(x_i)	Error err(x_i)
1.0000	0.5000000	0.5000000	0.0000000
1.1000	0.5839803	0.5787813	0.0051991
1.2000	0.6335731	0.6267660	0.0068070
1.3000	0.6625632	0.6558554	0.0067078
1.4000	0.6790560	0.6732069	0.0058491
1.5000	0.6879542	0.6832429	0.0047113
1.6000	0.6922892	0.6887536	0.0035356
1.7000	0.6939655	0.6915279	0.0024376
1.8000	0.6941922	0.6927249	0.0014672
1.9000	0.6937401	0.6931004	0.0006397
2.0000	0.6931000	0.6931472	0.0000472

International Journal of Physics, Chemistry and Mathematics
Vol. 2, No. 1, March 2014, PP: 1 – 16
Available online at http://ijpcm.com/

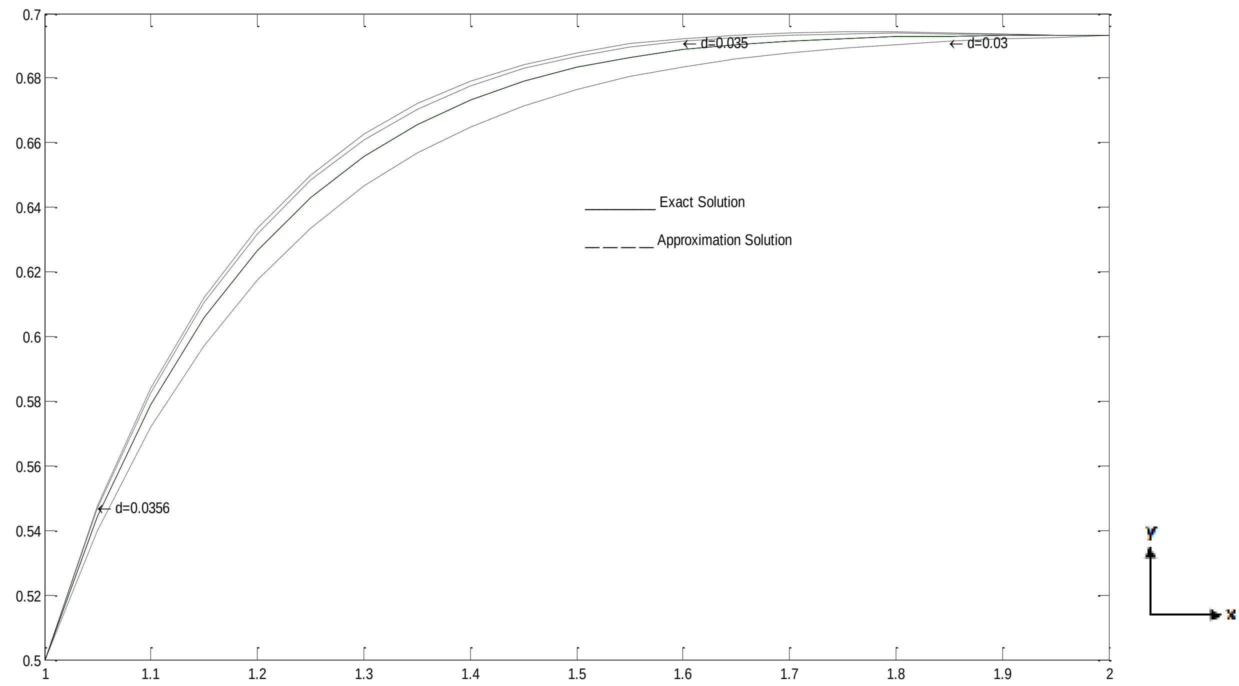


Figure 2: Computational results of Example 2 in graph

Conclusion

The method is an iterative on the deviating argument (δ). The process is to be repeated for different choices of δ until the approximate solution donot differ materially from the exact solution. The choice of δ will not be unique. By taking any value $0 < \delta \ll 1$ and adjusting the value of the deviating argument δ we can minimize the error.

Limitations

- If the deviating argument is becoming larger and larger (close and close to 1) the solution will deviate most from the exact solution.
- It doesn't show us the value of the deviating argument (δ). And it is not the same for all types of problems.
- It only works for Dirichlet "the first kind" boundary conditions.

Reference

[1] Chapara, S. C. (2001). Applied Numerical Methods with mathlab for Engineers and Scientists.Newyork,USA.

[2] Faires, R. L. (2005). Numerical Analysis.Boston,USA: Brooks/Cole.

[3] Jain, M. (1984). Numerical solution of Differential Equations. Delhi: Wiley Eastern Limited.

[4] Kiusalaas, J. (2005). Numerical integration Methods in Engineering with python. New York,USA: Cambridge University Press.

[5] Rao, G. (2003). Numerical Analysis. New Delhi,India: New Age International Publisher.

[6] Rosłoniec, S. (2008). Fundamental Numerical Methods. Poland: Springer.

[7] Taylor, G. M. (1996). Theory and Applications of Numerical Analysis. Scotland: Elsevier Science & Technology Books.