GANIT *J. Bangladesh Math. Soc.* (ISSN 1606-3694) 30 (2010) 51-58

NUMERICAL SOLUTIONS OF IVP USING FINITE ELEMENT METHOD WITH TAYLOR SERIES

Md. Shafiqul Islam [1], **Mostak Ahmed** [2] and **M. Alamgir Hossain** [3]

[1] *Department of Mathematics*
University of Dhaka, Dhaka-1000, Bangladesh
Email: mdshafiqul@yahoo.com

[2, 3] *Department of Mathematics,*
Jagannath University, Dhaka-1100, Bangladesh

Received 05.05.09 Accepted 25.04.11

ABSTRACT

In this paper, we use the integration technique together with the finite element method to approximate the numerical solution of an initial value problem of differential equations. The function of two variables is expanded into Taylor's series up to order two. We exploit Gauss-Legendre quadrature rules evaluating the integrals arising in the formulation of the present method to get the better accuracy.

1. Introduction

The initial value problem of ordinary differential equation is of the form [1, 2, 3]

$$y'(x) = f(x,y), \quad x \in [a,b] \tag{1}$$

with the initial condition

$$y(a) = y_0 \tag{2}$$

The above equations are equivalent to the integral equation

$$y(x) = y_0 + \int_a^x f(t,y)dt. \tag{3}$$

Instead of solving Equations (1) and (2), we will solve Equation (3). Thus, we have the formula

$$y_{m+1} = y_m + \int_{x_m}^{x_{m+1}} f(x,y)dx. \tag{4}$$

There are many ways approximating the integration of the equation (4). In this paper, by using the Taylor's series approximation, the function under the integration term $f(x,y)$ is to be transformed into the function of a single variable x, and then we use the Gauss-Legendre quadrature rule to evaluate the integral, which is the objective of this paper.

2. Formulation

We first discretize the interval $[a,b]$ into N equally spaced such that the mesh points are

$x_m = a + m\,h, \qquad \textit{for each } m = 0, 1, 2, ..., N.$

The common length of the step size is $h = (b-a)/N$. Now we map each subinterval into standard finite element $(-1 \le \xi \le 1)$, [4] as

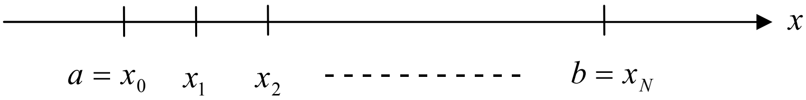

Figure 1: Discretization of the interval $[a,b]$

$$x(\xi) = x_1 L_1(\xi) + x_2 L_2(\xi) \tag{5}$$

where $L_1(\xi) = \frac{1}{2}(1-\xi)$ and $L_2(\xi) = \frac{1}{2}(1+\xi)$ (6)

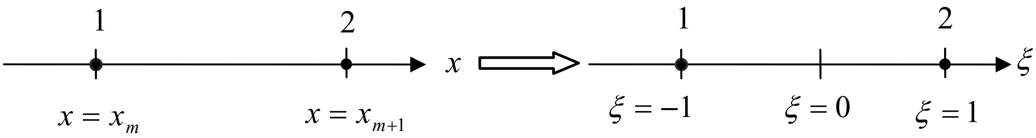

Figure 2: Transformation of arbitrary linear element into equivalent standard linear element

The Gauss Legendre quadrature rule [1, 2] is

$$\int_a^b f(x)dx \approx \sum_{k=1}^{n} w_k f(x_k) \tag{7}$$

where x_k and w_k are the Gauss points and the corresponding weight coefficients, respectively.

Now we transform the integral $\int_{x_m}^{x_{m+1}} f(x, y)dx$ in the right hand side of (4) into a standard integral using the transformation,

$$x(\xi) = \frac{1}{2}(x_{m+1} + x_m) + \frac{\xi}{2}(x_{m+1} - x_m) \tag{8}$$

$$dx = \frac{1}{2}(x_{m+1} - x_m)\; d\xi = \frac{h}{2}\; d\xi \tag{9}$$

Then $$\int_{x_m}^{x_{m+1}} f(x,y)dx = \frac{h}{2}\int_{-1}^{1} f(x(\xi), y(x(\xi)))\,d\xi \tag{10}$$

$$= \frac{h}{2}\sum_{i=1}^{n} f(x(\xi_i), y(x(\xi_i))) \quad \text{for } m = 0, 1, 2, ..., N. \tag{11}$$

For each *m*, $x(\xi_i) = x_m + s_i$ (say) and s_i's are constants for $i = 1, 2, ..., n.$

To determine $y(x(\xi_i))$, we use Taylor's approximation of order two:

$$y(x(\xi_i)) = y(x_m + s_i) = y_m + s_i\, f(x_m, y_m) + \frac{s_i^{\,2}}{2}[f_x(x_m, y_m) + f(x_m, y_m) f_y(x_m, y_m)] \tag{12}$$

where $y_m = y(x_m)$ and $i = 1, 2, ..., n.$

The equation (4) becomes to

$$y_{m+1} = y_m + \frac{h}{2}\int_{-1}^{1} f(x(\xi), y(x(\xi)))\, d\xi$$

$$= y_m + \frac{h}{2}\sum_{i=1}^{n} f(x(\xi_i), y(x(\xi_i))) = y_m + \frac{h}{2}\sum_{i=1}^{n} w_i F(\xi_i) \tag{13}$$

where $F(\xi) = f(x(\xi), y(x(\xi)))$ (14)

3. Algorithm and Program

To approximate the solution of the initial-value problem

$y' = f(x, y), \qquad a \le x \le b, \qquad y(a) = y_0,$

at $(N+1)$ equally spaced numbers in the interval $[a, b]$;

INPUT endpoints a, b; initial condition y_0; order of Gauss-Legendre Quadrature n

OUTPUT for some values of x to approximate the corresponding values of y.

Step 1 Set $N = (b - a)/h$;

$y_1 = y_0$;

Step2 For $m = 1, 2, 3, \cdots \cdots \cdots, N+1$ do Step 3

Step 3 Set $x_m = a + (m-1)h$;

Step 4 For $i = 1, 2, 3, \cdots \cdots \cdots, n$ do Step 5

Step 5 Set $\xi_i =$ gaussian points

$w_i =$ Corresponding weight coefficients

Step 6 For $m = 1, 2, 3, \cdots \cdots \cdots, N$ do Steps 7-11

Step 7 Set $sum = 0$;

Step 8 For $k = 1, 2, 3, \cdots \cdots \cdots, n$ do Steps 9-10

Step 9 Set $x\xi = \frac{1}{2}(x_{m+1} + x_m) + \frac{\xi_k}{2}(x_{m+1} - x_m)$;

$s = x\xi - x_m$;

$$y\xi = x_m + s\,f(x_m, y_m) + \frac{s^2}{2}[f_x(x_m, y_m) + f(x_m, y_m) f_y(x_m, y_m)];$$

Step 10 Set $sum = sum + w_k f(x\xi, y\xi)$;

Step 11 Set $y_{m+1} = y_m + \frac{h}{2}\,sum$;

Step 12 OUTPUT (x_i, y_i).

Step 13 STOP

MathematicaProgram

```
In[1]:= f[x_, y_] := x y^3/Sqrt[1 + x^2];
        DSolve[{y'[x] == x (y[x])^3/Sqrt[1 + x^2], y[0] == 1}, y[x], x]

Out[2]= {{y[x] -> 1/Sqrt[3 - 2 Sqrt[1 + x^2]]}}

In[3]:= << NumericalMath`GaussianQuadrature`

In[4]:= fx[x_, y_] = D[f[x, y], x];
        fy[x_, y_] = D[f[x, y], y];
        y[x_] := 1/Sqrt[3 - 2 Sqrt[1 + x^2]];

In[7]:= h = Input["h=?"];
        a = 0; b = 1; y0 = 1; N1 = (b - a)/h;
        Y = Table[y0, {i, 1, N1 + 1}];
        X = Table[a + (m - 1) h, {m, 1, N1 + 1}];
        n = Input["Order of the Gaussian Quadrature:"];
        ξ = Table[0, {i, 1, n}];
        W = Table[0, {i, 1, n}];
        GQ = GaussianQuadratureWeights[n, -1, 1];
```

```
Do[{ξ[[i]] = GQ[[i,1]];
   W[[i]] = GQ[[i,2]];}, {i, 1, n}];

In[20]:= Do[{sum = 0;
   Do[{
     xξ = 1/2 (X[[m+1]] + X[[m]]) + ξ[[k]]/2 (X[[m+1]] - X[[m]]);
     s = xξ - X[[m]];
     yξ = Y[[m]] + s f[X[[m]], Y[[m]]] + s^2/2 (fx[X[[m]], Y[[m]]] + f[X[[m]], Y[[m]]] fy[X[[m]], Y[[m]]]);
     sum = sum + W[[k]] f[xξ, yξ];}, {k, 1, n}]; Y[[m+1]] = Y[[m]] + h/2 sum}, {m, 1, N1}]
Print["Order of Gauss-Legengre Quadreture Rule=", n]
Print["Step Size=", h]
TableForm[
 Table[{SetPrecision[X[[i]], 2], SetPrecision[Y[[i]], 15], SetPrecision[y[X[[i]]], 15],
   ScientificForm[Abs[Y[[i]] - y[X[[i]]]], 10]}, {i, 1, N1 + 1, N1/10}],
 TableHeadings -> {None, {"   Xi", "      Yi", "    Exact", "    Error"}}]
```

4. Numerical Examples

In this section we consider two examples in which Example 1 is taken from [3] and Example 2 is taken from [2] to verify the effectiveness of our formulation presented in the previous section.

Example 1: Consider the numerical solution of the equation [3]

$$y'(x) = \frac{xy^3}{\sqrt{1+x^2}}, \qquad x \in [0,\ 1], \qquad y(0) = 1 \tag{15}$$

The analytic solution of the above equations is

$$y(x) = \frac{1}{\sqrt{3 - 2\sqrt{1+x^2}}}. \tag{16}$$

Example 2: Consider the numerical solution of the equation [2]

$$y'(x) = (x + 2x^3)y^3 - xy, \qquad 0 \le x \le 1, \qquad y(0) = \frac{1}{3} \tag{17}$$

The analytic solution of the above equations is

$$y(x) = (3 + 2x^2 + 6e^{x^2})^{-\frac{1}{2}} \tag{18}$$

Using present formulation for the third order Gauss-Legendre quadrature rule, the approximate solutions are shown in Table 1, which are obtained by the *Mathematica*

Program. We also compute the errors with exact solutions by $E = |Exact - Approx.|$ and are displayed in Table 1.

Table 1: Evaluation of approximate solutions using present formulation

	Example 1		Example 2	
x	(Approx. value)	Error (E)	(Approx. value)	Error (E)
Step size, $h = 0.1$				
0	1.00000000000000	0	0.333333333333333	0
0.10	1.00502506250022	1.261810898×10^{-7}	0.331856160086815	4.766246897×10^{-9}
0.20	1.02041042662246	1.883238168×10^{-6}	0.327475502187055	6.415010012×10^{-8}
0.30	1.04716212300636	7.842056093×10^{-6}	0.320337922687961	2.176400152×10^{-7}
0.40	1.08723242868709	2.338834206×10^{-5}	0.310669391497879	4.797366930×10^{-7}
0.50	1.14406142719502	6.137844035×10^{-5}	0.298752615848323	8.408279858×10^{-7}
0.60	1.22371468395705	1.561276345×10^{-4}	0.284902989428085	1.276412350×10^{-6}
0.70	1.33745993448850	4.133684514×10^{-4}	0.269447658788177	1.758875964×10^{-6}
0.80	1.50846807007518	1.234117332×10^{-3}	0.252710609020833	2.266745158×10^{-6}
0.90	1.79342317878687	4.733201783×10^{-3}	0.235004691372392	2.789187064×10^{-6}
1.0	2.38264733205191	3.156623032×10^{-2}	0.216629746691091	3.325897822×10^{-6}
Step size, $h = 0.01$				
0	1.00000000000000	0	0.333333333333333	0
0.10	1.00502518842526	$2.560505141\times10^{-10}$	0.331856155329971	$9.403477996\times10^{-12}$
0.20	1.02041230760477	2.255855724×10^{-9}	0.327475438108880	$7.192518803\times10^{-11}$
0.30	1.04716995613124	8.931210882×10^{-9}	0.320337705272432	$2.244858743\times10^{-10}$
0.40	1.08725579035485	2.667429633×10^{-8}	0.310668912240318	$4.791320118\times10^{-10}$
0.50	1.14412273431291	7.132245705×10^{-8}	0.298751775845289	$8.249514960\times10^{-10}$
0.60	1.22387062449678	1.870947743×10^{-7}	0.284901714253402	1.237666247×10^{-9}
0.70	1.33787278420842	5.187314858×10^{-7}	0.269445901603319	1.691106311×10^{-9}
0.80	1.50970052066639	1.666740491×10^{-6}	0.252708344441461	2.165785051×10^{-9}
0.90	1.79814905119406	7.329376235×10^{-6}	0.235001904837957	2.652628228×10^{-9}
1.0	2.41414427510185	6.928727125×10^{-5}	0.216626423945401	3.152131417×10^{-9}
Step size, $h = 0.001$				
0	1.00000000000000	0	0.333333333333333	0
0.10	1.00502518868105	$2.584599201\times10^{-13}$	0.331856155320577	$9.270362256\times10^{-15}$
0.20	1.02041230985836	$2.269961996\times10^{-12}$	0.327475438037027	$7.188694084\times10^{-14}$
0.30	1.04716996505345	$9.000133971\times10^{-12}$	0.320337705048170	$2.237099395\times10^{-13}$
0.40	1.08725581700220	$2.694355850\times10^{-11}$	0.310668911761663	$4.776734563\times10^{-13}$
0.50	1.14412280556312	$7.224887355\times10^{-11}$	0.298751775021159	$8.217870828\times10^{-13}$
0.60	1.22387081140135	$1.902009661\times10^{-10}$	0.284901713016968	$1.232070002\times10^{-12}$
0.70	1.33787330241000	$5.298983474\times10^{-10}$	0.269445899913896	$1.682931572\times10^{-12}$
0.80	1.50970218569168	1.715205755×10^{-9}	0.252708342277830	$2.154609824\times10^{-12}$
0.90	1.79815637292522	7.645072131×10^{-9}	0.235001902187967	$2.638028684\times10^{-12}$
1.0	2.41421348748317	7.488992804×10^{-8}	0.216626420796404	$3.133854287\times10^{-12}$

Step size, $h = 0.0001$				
0	1.00000000000000	0	0.333333333333333	0
0.10	1.00502518868131	$1.332267630\times10^{-15}$	0.331856155320568	$3.885780586\times10^{-16}$
0.20	1.02041230986062	$5.995204333\times10^{-15}$	0.327475438036956	$9.992007222\times10^{-16}$
0.30	1.04716996506244	$1.287858709\times10^{-14}$	0.320337705047948	$1.332267630\times10^{-15}$
0.40	1.08725581702911	$3.308464613\times10^{-14}$	0.310668911761187	$1.665334537\times10^{-15}$
0.50	1.14412280563529	$7.549516567\times10^{-14}$	0.298751775020339	$1.554312234\times10^{-15}$
0.60	1.22387081159136	$1.949551631\times10^{-13}$	0.284901713015738	$2.331468352\times10^{-15}$
0.70	1.33787330293937	$5.342393194\times10^{-13}$	0.269445899912215	$2.164934898\times10^{-15}$
0.80	1.50970218740516	$1.726396803\times10^{-12}$	0.252708342275678	$2.720046410\times10^{-15}$
0.90	1.79815638056261	$7.684519687\times10^{-12}$	0.235001902185332	$3.330669074\times10^{-15}$
1.0	2.41421356229761	$7.548761616\times10^{-11}$	0.216626420793273	$3.552713679\times10^{-15}$

Now we compare our results with the results in the literature [3] which have been obtained by using various Newton-Cote's formulas. Note that, Podisuk *et al* [3] strongly recommended that the formulas (6, 9, 12, 15, 18, 21, 24, 27 and 30) should be used to find the numerical solution of an IVP. In contrast, we observe that our present formula gives more accurate results.

Table 2: Comparison of the present errors with the errors of Reference [3]

Formula of Reference [3]	$x = 0.1$, $h = 0.1$	$x = 0.1$, $h = 0.0001$	$x = 1.0$, $h = 0.0001$
6	$1.2676962557\times10^{-5}$	$1.8189894035\times10^{-12}$	$1.2563396012\times10^{-7}$
9	$2.4998819848\times10^{-5}$	$3.0922819860\times10^{-11}$	$1.9134677132\times10^{-7}$
12	$8.4965516520\times10^{-6}$	$8.5544904493\times10^{-12}$	$9.0301900950\times10^{-8}$
15	$2.9294220072\times10^{-6}$	$5.4569682106\times10^{-12}$	$2.0128936740\times10^{-8}$
18	$2.6840044534\times10^{-6}$	$5.4569682106\times10^{-12}$	$2.0114384824\times10^{-8}$
21	$1.1836842218\times10^{-6}$	$5.4569682106\times10^{-12}$	$2.0128936740\times10^{-8}$
24	$1.3301713442\times10^{-7}$	$5.4569682106\times10^{-12}$	$2.0128936740\times10^{-8}$
27	$1.2270877692\times10^{-7}$	$5.4569682106\times10^{-12}$	$2.0125298561\times10^{-8}$
30	$1.3093116769\times10^{-7}$	$5.4569682106\times10^{-12}$	$2.0121660782\times10^{-8}$
Error in Present Method			
	$1.2618108980\times10^{-7}$	$1.3322676296\times10^{-15}$	$7.5487616158\times10^{-11}$

5. Conclusion

Using integration method and finite element method for solving IVP we observe that the error terms are almost smaller than that of [3]. For simplicity of calculation we use first three terms of Taylor's series, but if we use more terms we can find more accurate results. On the other hand, to perform the numerical integration using finite element method we use two-point or three-point Gaussian quadrature, but if we use more Gaussian quadrature points, we also find more accurate results.

REFERENCES

1. Philip J. Davis, Philip Robinowitz, Methods of Numerical Integration, Dover publications Inc., USA, 2007.
2. R.L. Burden, J. Douglas Faires, Numerical Analysis, Seventh Ed., Thomson Learning, 2001.
3. Maitree Podisuk, Ungsana Chundang, Wannaporn Sanprasert. Single Step Formulas and Multi-step Formulas of the Integration Method for Solving the Initial Value Problem of Ordinary Differential Equation, Applied Mathematics and Computation, 190, (2007) 1438 – 1444.
4. P.E. Lewis and J.P. Ward – The Finite Element Method: Principles and Applications, Addison-Wesley Publishing Company, 1991.