Matematika, 2002, Jilid 18, Bil. 2, hlm. 79–90

Numerical Treatment of Delay Differential Equations by Runge-Kutta Method Using Hermite Interpolation

Fudziah Ismail, Raed Ali Al-Khasawneh,
Aung San Lwin & Mohamed Suleiman
Department of Mathematics,
Universiti Putra Malaysia,
43400 UPM, Serdang, Malaysia.

Abstract Embedded Diagonally Implicit Runge-Kutta methods of different orders are used for the treatment of delay differential equations. The delay argument is approximated using an appropriate Hermite Interpolation. The numerical results based on these methods are compared and the Q-stability region of the methods are presented.

1 Introduction

In the last two decades there has been a growing interest in the numerical treatment of delay differential equations (DDEs). This is due to their versatility in the mathematical modeling of processes in various application fields. DDEs generally provide the best and sometimes the only realistic simulation of observed phenomena. Such work can be found in Orbele and Pesch (1981), Baker and Paul (1992) and Zennaro (1988). First order DDE can be written as

$$\begin{aligned} y'(t) &= f(t, y, y(t-\tau)) \quad t > t_0 \\ y(t) &= \varphi(t) \quad t \leq t_0 \end{aligned} \tag{1}$$

if it has one delay term only, or

$$\begin{aligned} y'(t) &= f(t, y, y(t-\tau_1), \ldots y(t-\tau_n)) \quad t > t_0 \\ y(t) &= \varphi(t) \quad t \leq t_0 \end{aligned} \tag{2}$$

if it has more than one delay term.

$\varphi(t)$ is the initial function, $\tau(t, y(t))$ is called the delay, is called the delay argument, the value of $y(t - \tau(t, y(t))$ is the solution of the delay term or commonly referred to as the delay term only. If the delay is a constant then it is called constant delay, if it is function of time t, then it is called time dependent delay, if it is a function of time t and $y(t)$ then it is called the state dependent delay.

2 Numerical Treatment Of DDEs

Most numerical methods for solving ordinary differential equation of the form

$$y'(t) = f(t, y(t)) \quad y(t_0) = c \tag{3}$$

can be adapted to solve DDEs. The range comprises of one step methods such as Runge-Kutta method and Euler method, multistep and also block implicit method. When a q-stage Singly Diagonally Implicit Runge-Kutta (SDIRK) method is used to solve (3) at the point t_{n+1} , the following equations will be obtained:

$$\begin{aligned}
k_1 &= f(t_n + c_1 h, y_n + h a_{11} k_1) \\
k_2 &= f(t_n + c_2 h, y_n + h a_{21} k_1 + h a_{22} k_2) \\
&\vdots \\
k_i &= f(t_n + c_i h, y_n + h \sum_{j=1}^{i} a_{ij} k_j) \\
y_{n+1} &= y_n + h \sum_{i=1}^{q} b_i k_i \quad (i = 1, \ldots, q)
\end{aligned}$$

$y_n + h \sum_{j=1}^{i} a_{ij} k_j$ is called the internal stage of the method.

When it is adapted to DDE(1) we have

$$\begin{aligned}
k_i &= f(t_n + c_i h, y_n + h \sum_{j=1}^{i} a_{ij} k_j, y(t_n + c_i h - \tau)) \\
y_{n+1} &= y_n + h \sum_{i=1}^{q} b_i k_i
\end{aligned}$$

$y(t_n + c_i h - \tau)$ is the delay term and interpolation is needed to approximate the value. There are a number of techniques for obtaining the approximations which has been discussed in Neves (1981), In't Hout (1992) and Karoui (1992). In this paper Hermite interpolation is used to approximate the delay term. The interpolation order and hence the number of support points have to be adapted to the order of the method.

If p denotes the order of the Runge-Kutta method used, the interpolation order q must be greater or equal to p. Let i_p denotes the number of support points for Hermite interpolation then

$$2i_p > p \tag{4}$$

In this paper fifth order embedded (SDIRK) method (4,5) in (5,6) (see Ismail and Suleiman (1999)) and fourth order embedded SDIRK (3,4) in (4,6) due to Hairer and Wanner (1991) were used to solve DDEs.

For the fifth order method three support points were used, so that equation (4) holds and for the fourth order method we are also using three support points since if two support points were used the order of the interpolations is less than the order of the method itself.

In the numerical treatment of delay differential equations two essential difficulties occur. First is the evaluation of the delay term which has been discussed earlier and secondly is the jump discontinuities of the solution in various derivatives, which usually are the characteristic of DDEs.

When solving DDEs, one of the basic requirements is the storage of sufficient back information so that the method can evaluate the delay term when it is required at some point $t \leq t_n$. The amount of information to be stored at each time step depends on the method for approximating the delay term, but the interval on which the information is to be stored and the quantities to be stored on that interval should be flexible and adaptable for each problem, depending on the nature of the delay term and accuracy required. If the delay term falls at some point $t \leq t_0$, then the initial function must be used.

The delay argument may fall in the current step because it is smaller than the stepsize or may even vanish, we call this type of delay a small delay or when the delay vanishes we call it vanishing delay. These types of delays are handled by either restricting the stepsize to be smaller than the delays or using extrapolation.

Discontinuities are handled by the error control mechanism or locating the discontinuity points by solving associated nonlinear equations. In this paper the test equations used have the analytical solution so we knew the discontinuity point hence make it a mesh point so that the stepsize does not cross the discontinuity point.

The main concern here is DDE which has a single delay, both time dependent and state dependent and of the large type.

3 Stability Of The Method

There are many concepts of stability of numerical methods when applied to DDE, depending on the test equation as well as the delay term involved see Bellen et al., (1988), Tian Hongjiang and Kuang Jiaoxun (1995) and Guang-Da Hu and Meguid, S. A. (1999). One of the most commonly used test equation in the literature is,

$$\begin{aligned} y'(t) &= \lambda y(t) + \mu y(t-\tau), \quad t \geq t_0 \\ y'(t) &= \varphi(t), \quad t \leq t_0 \end{aligned} \tag{5}$$

λ and $\mu \in C$, $\tau > 0$ and φ is continuous.

If $\lambda = 0$, the following equation is obtained

$$\begin{aligned} y'(t) &= \mu y(t-\tau) \quad t \geq t_0 \\ y'(t) &= \varphi(t) \quad t \leq t_0 \end{aligned} \tag{6}$$

Barwell (1975) introduced the following concept of stability.

Definition: Given a numerical method for DDEs, the P-stability region of the method is the set S_p of pairs of (α, β), such that the numerical solution of (5) asymptotically vanishes for steplengths h satisfying

$$h = \frac{\tau}{m}, \quad m \quad \text{is a positive integer.}$$

Definition: If $\mu \in C$, Q-stability region of the method is the set S_Q of β, such that the numerical solution of (6) asymptotically vanishes for steplengths h satisfying

$$h = \frac{\tau}{m}, \quad m \quad \text{is a positive integer.}$$

$\alpha = h\lambda$ and $\beta = h\mu$.

When q-stage Runge-Kutta method is applied to DDE (5), using Hermite interpolation for the delay term, the following equations are obtained.

$$\begin{aligned} k_{n+1}^{(i)} &= f(t_n + c_i h, y_n + h\sum_{j=1}^{i} a_{ij} k_{n+1}^{(j)} + \sum H(c_i) y_{n-m+l} + \bar{H}(c_i) h y'_{n-m+l}) \\ y_{n+1} &= y_n + h\sum_{i=1}^{q} b_i k_{n+1}^{(i)} \end{aligned} \qquad (7)$$

Where H and $\bar{H}$ are the coefficients of Hermite interpolation.

Define $u = (1, \ldots, 1)^T$ and for $n \geq 1$

$$\begin{aligned} \underline{\mathrm{k}}_n &= (k_n^{(1)}, \ldots, \underline{\mathrm{k}}_n^{(q)})^T \\ \underline{\mathrm{b}} &= (b_1, b_2, \ldots, b_q)^T \\ H_l(\underline{\mathrm{c}}) &= H_l(c_1), \ldots, H_l(c_q) \\ \bar{H}_l(\underline{\mathrm{c}}) &= \bar{H}_l(c_1), \ldots, \bar{H}_l(c_q) \end{aligned}$$

For $n \geq m$ (7) takes the form

$$\underline{\mathrm{k}}_{n+1} = \lambda(y_n u + hA\underline{\mathrm{k}}_{n+1}) + \mu\sum_{l=r}^{s}(H_l(\underline{\mathrm{c}}) y_{n-m+l} + h\bar{H}_l(\underline{\mathrm{c}}) y_{n-m+l} + h\bar{H}_l(\underline{\mathrm{c}}) y'_{n-m+l}) \qquad (8)$$

$$y_{n+1} = y_n + hb^T \underline{\mathrm{k}}_{n+1} \qquad (9)$$

Replacing y'_{n-m+l} by $\lambda y_{n-m+l} + \mu y_{n-2m+l}$, we will have

$$\underline{\mathrm{k}}_{n+1} = \lambda y u + h\lambda A\underline{\mathrm{k}}_{n+1} + \mu\sum_{l=r}^{s} H_l(\underline{\mathrm{c}}) y_{n-m+l} + \mu h\lambda\sum_{l=r}^{s} \bar{H}_l(\underline{\mathrm{c}}) y_{n-m+l} + h\mu^2\sum_{l=r}^{s} \bar{H}_l(\underline{\mathrm{c}}) y_{n-2m+l}$$

hence, we have

$$(I - h\lambda A)\underline{\mathrm{k}}_{n+1} = \lambda y_n u + \mu\sum_{l=r}^{s}\left[(H_l(\underline{\mathrm{c}}) + h\lambda\bar{H}_l(\underline{\mathrm{c}}c)) y_{n-m+l} + h\mu\bar{H}(\underline{\mathrm{c}})_l y_{n-2m+l}\right]$$

$$\begin{aligned} h\underline{\mathrm{k}}_{n+1} &= h\lambda(I-h\lambda A)^{-1}y_n u + \\ & \quad h\mu(I-h\lambda A)^{-1}\sum_{l=r}^{s}\left[(H_l(\underline{\mathrm{c}})+h\lambda\bar{H}_l(\underline{\mathrm{c}}))y_{n-m+l}+h\mu\bar{H}(\underline{\mathrm{c}})_l y_{n-2m+l}\right] \qquad (10) \end{aligned}$$

replacing the above equation into (9) and taking $\alpha = h\lambda$, $\beta = h\mu$ and $\eta = (I - h\lambda A)^{-1}$ we have

$$y_{n+1} = y_n + \alpha b^T\eta y_n u + \beta b^T\eta\sum_{l=r}^{s}\left[(H_l(\underline{\mathrm{c}})+h\lambda\bar{H}_l(\underline{\mathrm{c}}))y_{n-m+l}+h\mu\bar{H}(\underline{\mathrm{c}})_l y_{n-2m+l}\right] \qquad (11)$$

Rewriting

$$Y_{n+1} = \begin{pmatrix} y_{n+1} \\ h\underline{\mathrm{k}}_{n+1} \end{pmatrix},$$

equation (10) can be written as

$$\begin{pmatrix} 1 & \underline{0} \\ \underline{0} & I \end{pmatrix} Y_{n+1} = \begin{pmatrix} 1+\alpha b^T\eta u & \underline{0} \\ \alpha\eta u & \underline{\mathrm{u}} \end{pmatrix} Y_n + \begin{pmatrix} \beta b^T\eta\sum H_l(\underline{\mathrm{c}})+\lambda\bar{H}_l(\underline{\mathrm{c}})) & \underline{0} \\ \beta\eta\sum H_l(\underline{\mathrm{c}})+\alpha\bar{H}_l(\underline{\mathrm{c}})) & \underline{0} \end{pmatrix} Y_{n-m+l} +$$

$$\begin{pmatrix} \beta^2 b^T\eta\sum\bar{H}_l(\underline{\mathrm{c}}) & \underline{0} \\ \beta^2\eta\sum\bar{H}_l(\underline{\mathrm{c}}) & \underline{0} \end{pmatrix} Y_{n-2m+l}$$

I is the identity matrix, replacing Y by ζ, the stability polynomial of the method is

$$S(\alpha,\beta,\zeta) = \det[\begin{pmatrix} 1 & \underline{0} \\ \underline{0} & I \end{pmatrix}\zeta^{n+1} - \begin{pmatrix} 1+\alpha b^T\eta u & \underline{0} \\ \alpha\eta u & \underline{\mathrm{u}} \end{pmatrix}\zeta^n - \begin{pmatrix} \beta b^T\eta\sum H_l(\underline{\mathrm{c}})+\lambda\bar{H}_l(\underline{\mathrm{c}})) & \underline{0} \\ \beta\eta\sum H_l(\underline{\mathrm{c}})+\alpha\bar{H}_l(\underline{\mathrm{c}})) & \underline{0} \end{pmatrix}\zeta^{n-m+l}$$

$$-\begin{pmatrix} \beta^2 b^T\eta\sum\bar{H}_l(\underline{\mathrm{c}}) & \underline{0} \\ \beta^2\eta\sum\bar{H}_l(\underline{\mathrm{c}}) & \underline{0} \end{pmatrix}\zeta^{n-2m+l}]$$

Hence

$$\begin{aligned} S(\alpha,\beta,\zeta) &= \zeta^{n+1} - (1+\alpha b^T\eta u)\zeta^n - \beta b^T\eta\sum_l(H_l(\underline{\mathrm{c}})+\lambda\bar{H}_l(\underline{\mathrm{c}}))\zeta^{n-m+l} \\ &- \beta^2 b^T\eta\sum_l\bar{H}_l(\underline{\mathrm{c}})\zeta^{n-2m+l} \qquad (12) \end{aligned}$$

The above polynomial is called the P-stability polynomial of the method using Hermite interpolation to approximate the delay term. If $\lambda = 0$, hence $\alpha = 0$, then we obtain

$$S(\beta,\zeta) = \zeta^{n+1} - \zeta^n - \beta b^T\eta\sum_l(H_l(\underline{\mathrm{c}}))\zeta^{n-m+l} - \beta^2 b^T\eta\sum_l\bar{H}_l(\underline{\mathrm{c}})\zeta^{n-2m+l} \qquad (13)$$

that is the Q-stability polynomial of the method.

By taking $n-2m-1=0$, so that the lowest order of ζ is zero and using three points for the interpolations for both the methods we have three values of y and y', that is six data altogether so that the interpolations are at least of the same order as the method.

Hermite polynomial of least degree agreeing with y and y' at $t_0, \ldots, t_n$ is a polynomial of degree at most $2n+1$ given by

$$P_{2n+1}(t) = \sum_{j=0}^{n} H_{n,j}(t)y(t_j) + \sum_{j=0}^{n} \bar{H}_{n,j}(t)y'(t_j)$$

where

$$\begin{aligned} H_{n,j}(t) &= [1-2(t-t_j)L'_{n,j}(t_j)]L^2_{n,j}(t) \\ \bar{H}_{n,j}(t) &= (t-t_j)L^2_{n,j}(t) \end{aligned}$$

and

$$L_j(t) = \Pi_{l=r}^{s} \frac{(t-l)}{(j-l)} \quad (j \neq l)$$

In this case, the coefficients of the 3-point Hermite interpolation are

$$\begin{aligned} H_{2,0}(c) &= 0.25(4+3c)c^2(c-1)^2 \\ H_{2,1}(c) &= (c+1)^2c(c-1)^2 \\ H_{2,2}(c) &= 0.25(4-3c)c^2(c+1)^2 \\ \bar{H}_{2,0}(c) &= 0.25(1+c)c^2(c-1)^2 \\ \bar{H}_{2,1}(c) &= c(1+c)(c-1)^2 \\ \bar{H}_{2,2}(c) &= 0.25(c-1)c^2(c+1)^2 \end{aligned}$$

And taking $H_l = H_{2,j}$ and $\bar{H}_l = \bar{H}_{2,j}$, $h=1$ and $m=1$, equation (13) becomes

$$S(\beta,\zeta) = \zeta^4 - \zeta^3 - \beta b^T \eta \sum_{l=1}^{3} (H_l(\underline{c}))\zeta^l - \beta^2 b^T \eta \sum_{l=0}^{2} \bar{H}_l(\underline{c})\zeta^l \qquad (14)$$

The Q-stability region of the method is the region where all the roots of the polynomial (ζ) is less or equal to 1, hence substituting the value of $\zeta = \cos\theta + i\sin\theta$, and solve the equation for $\beta = a+ib$ we obtained the required regions. Figure 1 below gives the Q-stability regions for both methods, where the regions lie inside the loop.

4 Numerical Experiment

Below are some of the problems tested, which were obtained from Al-Mutib (1977).

Problem

No 1:-

$$\begin{aligned} y'(t) &= -y\exp(1-\frac{1}{t})+1, \quad 1 \le t \le 10 \\ y(t) &= \ln(t), \quad 0 \le t \le 1 \\ \tau(t) &= t-\exp(1-\frac{1}{t}) \end{aligned}$$

Solution: $y(t) = \ln(t), \quad t \in [0, e]$

No 2:-

$$\begin{aligned} y'(t) &= -y(t-1+e^{-t})+\sin(t-1+e^{-t})+\cos(t), \quad t \ge 0 \\ y(t) &= \sin(t), \quad t \le 0 \\ \tau(t) &= 1-e^{-t} \end{aligned}$$

Solution: $y(t) = \sin(t), \quad t \in [0, 10]$

No 3:-

$$\begin{aligned} y'(t) &= \cos(t)y(y(t)-2), \quad t \ge 0 \\ y(t) &= 1, \quad t \le 0 \\ \tau(t) &= t-y(t)+2 \end{aligned}$$

Solution: $y(t) = \sin(t)+1, \quad t \in [0, 10]$

No 4:-

$$\begin{aligned} y'(t) &= \frac{1}{2}\exp(y(y(t)-\ln 2+1)), \quad 1 \le t \le 3 \\ y(t) &= 0, \quad t \le 1 \\ \tau(t) &= t-y(t)+\ln 2-1 \end{aligned}$$

Solution:

$$y(t) = \begin{cases} \ln(t) & 1 \le t \le 2 \\ \frac{t}{2}+\ln 2-1 & 2 \le t \le 3 \end{cases} \qquad t \in [1, 3]$$

No 5:-

$$\begin{aligned} y'(t) &= \frac{1}{t}y(t)y(\ln(y(t))), \quad t \ge 1 \\ y(t) &= 1, \quad t \le 1 \\ \tau(t) &= t-\ln(y(t)) \end{aligned}$$

Solution:

$$y(t) = \begin{cases} t & 1 \le t \le e \\ \exp(\frac{t}{e}) & e \le t \le e^2 \end{cases} \qquad t \in [1, e^2]$$

No 6:-

$$\begin{aligned}
y_1'(t) &= y_2(t), \quad 1 \le t \le 10 \\
y_2'(t) &= -y_2(\exp(1-y_2(t))^2 \exp(t-y_2(t)), \quad 1 \le t \le 10 \\
y_1(t) &= \ln(t), \quad 0 \le t \le 1 \\
y_2(t) &= \frac{1}{t}, \quad 0 \le t \le 1 \\
\tau(t) &= t - \exp(1-y_2(t))
\end{aligned}$$

Solution:

$$\begin{aligned}
y_1(t) &= \ln(t), \quad 0 \le t \le 10 \\
y_2(t) &= \frac{1}{t}, \quad 0 \le t \le 10
\end{aligned}$$

The numerical results are obtained when the problems are solved by fifth order SDIRK method F1(C) in Ismail and Suleiman and fourth order SDIRK method due to Hairer (1991) using 3-point Hermite interpolation to approximate the delay term and are given in Tables 1-6. The notations used are as follows:

TOL	The chosen tolerance.
FCN	The number of function evaluation.
STEP	The number of successful steps.
FSTEP	The number of failed steps.
MAX ERROR	Absolute value of the true solution minus the computed solution.

The notation $6.453225(-5)$ means $6.453225x10^{-5}$.

Method:

F1 Using SDIRK method (4,5) in (5,6)

F2 Using SDIRK method (3,4) in (4,5)

5 Conclusion and Discussion

From the numerical results, it was observed that, the fifth order method gives smaller number of function evaluation though the fifth order method has six stages meaning at every step, six functions evaluations has to be done compared to the five function evaluations for the fourth order method. In terms of number of steps, the fifth order method also gives smaller number of steps and also less number of failed or unsuccessful steps, but the fourth order method gives a slightly smaller error compared to the fifth order method, this is because the interpolation used is of order five, one order higher than the method itself. Overall it can be concluded that the fifth order method is superior compared to the fourth order method despite of the same order of interpolation used for the delay term.

Both methods though of different orders have almost the same region of Q-stability. The reason for this is that the test equation comprises only of the delay term and Hermite interpolations used for both methods are of the same order.

References

[1] Al-Mutib, A. N. (1977). *Numerical Methods for Solving Delay Differential Equations*, Ph.D Thesis, University of Manchester, United Kingdom.

[2] Baker, C. T. H and Paul, C. A. H. (1994). *Computing stability regions-Runge-Kutta methods for delay differential equations*, IMA Journal of Numerical Analysis, 14:347-362.

[3] Barwell, V. K. (1975). *Special stability problems for functional equations*, BIT, 15: 130-135.

[4] Bellen, A. Jackiewicz, Z. and Zennaro, M. (1988). *Stability Analysis of one-step method for neutral delay differential equations*, Numer. Math, 52:605-619.

[5] Guang Da-Hu and Meguid S. A (1999). *Stability of Runge-Kutta methods for delay differential systems with multiple delays*, IMA Journal of Numer Math, 19:349-356.

[6] Hairer, E. and Wanner, G. (1991). *Solving Ordinary Differential Equations II, Stiff and Differential Algebraic Problems*, Berlin: Springer-Verlag.

[7] In't Hout, K. J. (1992). *A new interpolation procedure for adapting Runge-Kutta methods to delay differential equations*, BIT, 32: 634-649.

[8] Ismail, F. and Suleiman, M. B. (1988). *Embedded Singly Diagonally Implicit Runge-Kutta method (4,5) in (5,6), for the integration of stiff systems of ODEs*, Int, J. Comput. Math, 66: 325-341.

[9] Karoui, A. (1992). *On the Numerical Solution of Delay Differential Equations*, Masters Thesis University of Ottawa, Canada.

[10] Neves, K. W. (1981). *Control of interpolatory error in retarded differential equations.* ACM, Trans. Math. Soft.,7: 421-444.

[11] Orbele, H. J. and Pesch, H. J (1981). *Numerical treatment of delay differential equations by Hermite interpolation*, Numer Math. 37:235-255.

[12] Tian Hongjiang and Kuang Jiaoxun (1995). *The stability -method in the numerical solution of delay differential equations with several delay terms*, Journal of Comput. Appl. Math. 10(1):171-181

[13] Zennaro, M. (1986). *The P-stability properties of Runge-Kutta methods for delay differential equations*, Numer Math, 49: 305-318.

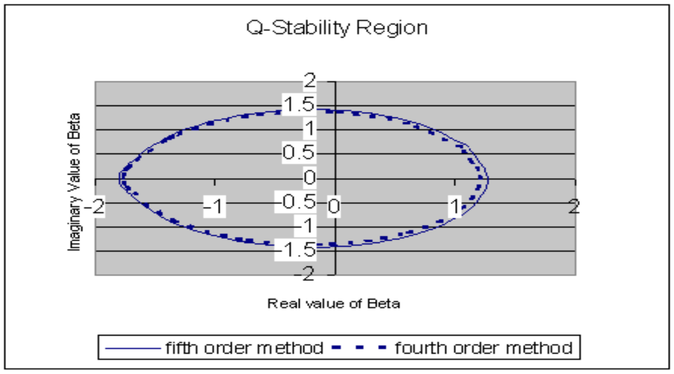
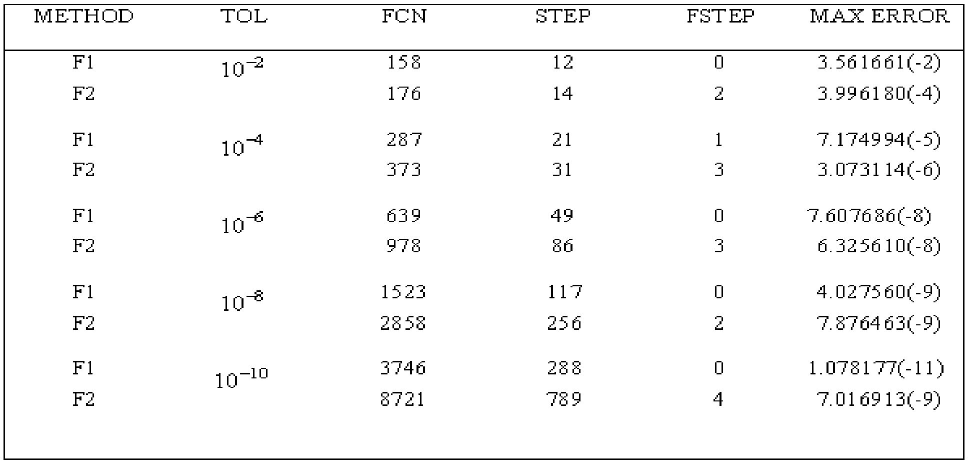


Figure 1:

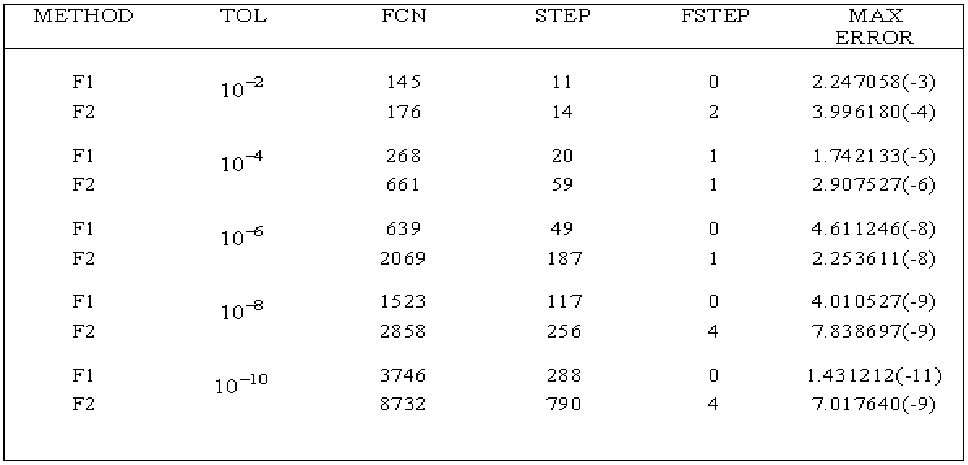
METHOD	TOL	FCN	STEP	FSTEP	MAX ERROR
F1	10^{-2}	119	9	0	1.500389(-2)
F2		101	11	0	1.971104(-2)
F1	10^{-4}	223	17	0	1.301600(-4)
F2		230	19	0	1.263141(-3)
F1	10^{-6}	444	34	0	9.068019(-6)
F2		518	46	0	3.693805(6)
F1	10^{-8}	977	75	0	2.451500(-7)
F2		1315	120	0	4.849865(-8)
F1	10^{-10}	2210	168	0	7.784948(-9)
F2		3845	349	0	1.296355(-8)

Table 1: Numerical Results for Problem 1



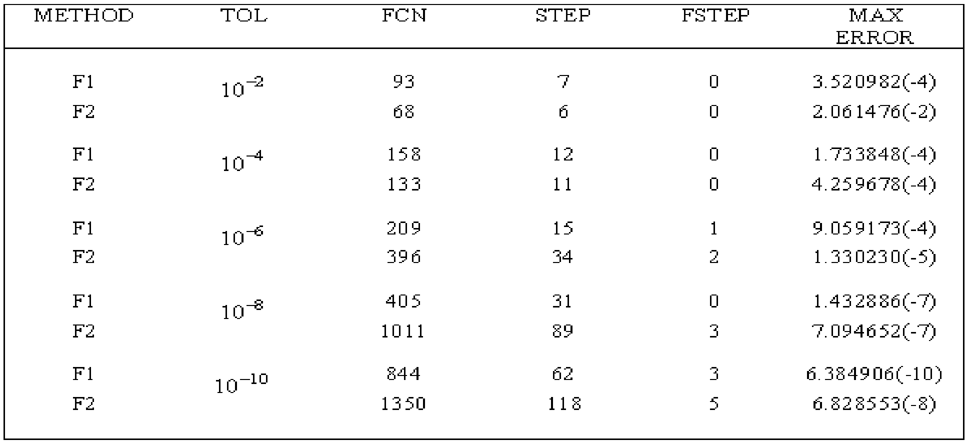
METHOD	TOL	FCN	STEP	FSTEP	MAX ERROR
F1	10^{-2}	158	12	0	3.561661(-2)
F2		176	14	2	3.996180(-4)
F1	10^{-4}	287	21	1	7.174994(-5)
F2		373	31	3	3.073114(-6)
F1	10^{-6}	639	49	0	7.607686(-8)
F2		978	86	3	6.325610(-8)
F1	10^{-8}	1523	117	0	4.027560(-9)
F2		2858	256	2	7.876463(-9)
F1	10^{-10}	3746	288	0	1.078177(-11)
F2		8721	789	4	7.016913(-9)

Table 2: Numerical Results for Problem 2



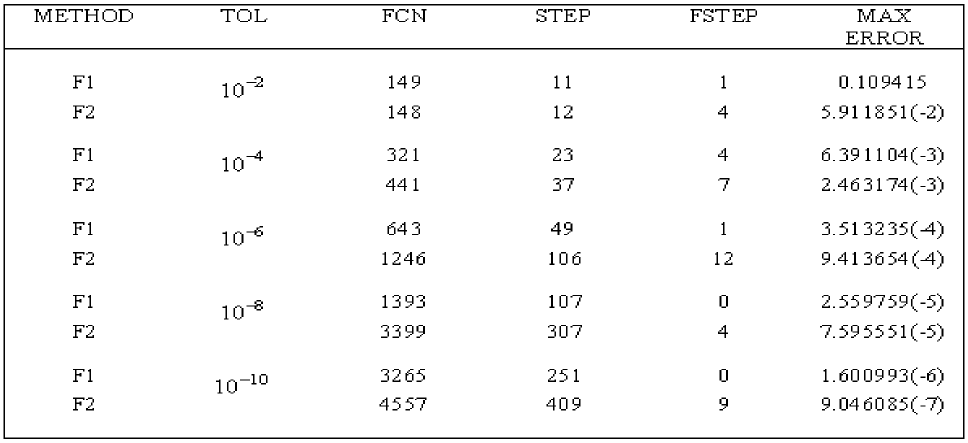
METHOD	TOL	FCN	STEP	FSTEP	MAX ERROR
F1	10^{-2}	145	11	0	2.247058(-3)
F2		176	14	2	3.996180(-4)
F1	10^{-4}	268	20	1	1.742133(-5)
F2		661	59	1	2.907527(-6)
F1	10^{-6}	639	49	0	4.611246(-8)
F2		2069	187	1	2.253611(-8)
F1	10^{-8}	1523	117	0	4.010527(-9)
F2		2858	256	4	7.838697(-9)
F1	10^{-10}	3746	288	0	1.431212(-11)
F2		8732	790	4	7.017640(-9)

Table 3: Numerical Results for Problem 3



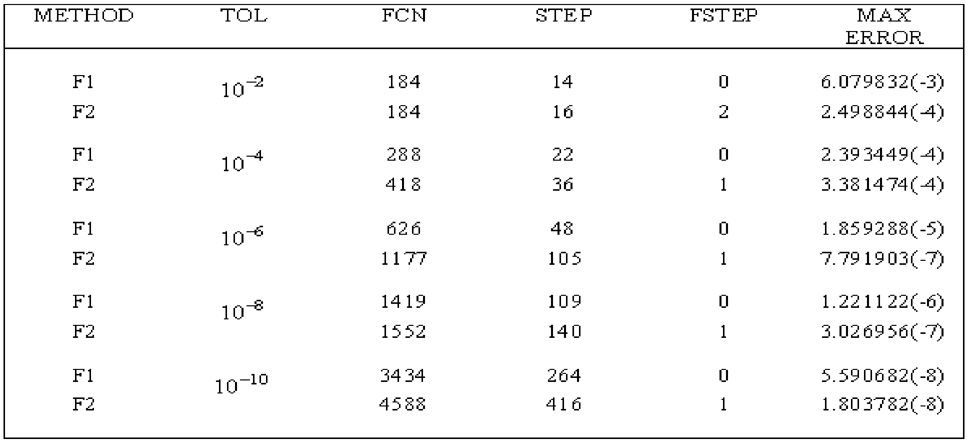
METHOD	TOL	FCN	STEP	FSTEP	MAX ERROR
F1	10^{-2}	93	7	0	3.520982(-4)
F2		68	6	0	2.061476(-2)
F1	10^{-4}	158	12	0	1.733848(-4)
F2		133	11	0	4.259678(-4)
F1	10^{-6}	209	15	1	9.059173(-4)
F2		396	34	2	1.330230(-5)
F1	10^{-8}	405	31	0	1.432886(-7)
F2		1011	89	3	7.094652(-7)
F1	10^{-10}	844	62	3	6.384906(-10)
F2		1350	118	5	6.828553(-8)

Table 4: Numerical Results for Problem 4



METHOD	TOL	FCN	STEP	FSTEP	MAX ERROR
F1	10^{-2}	149	11	1	0.109415
F2		148	12	4	5.911851(-2)
F1	10^{-4}	321	23	4	6.391104(-3)
F2		441	37	7	2.463174(-3)
F1	10^{-6}	643	49	1	3.513235(-4)
F2		1246	106	12	9.413654(-4)
F1	10^{-8}	1393	107	0	2.559759(-5)
F2		3399	307	4	7.595551(-5)
F1	10^{-10}	3265	251	0	1.600993(-6)
F2		4557	409	9	9.046085(-7)

Table 5: Numerical Results for Problem 5



METHOD	TOL	FCN	STEP	FSTEP	MAX ERROR
F1	10^{-2}	184	14	0	6.079832(-3)
F2		184	16	2	2.498844(-4)
F1	10^{-4}	288	22	0	2.393449(-4)
F2		418	36	1	3.381474(-4)
F1	10^{-6}	626	48	0	1.859288(-5)
F2		1177	105	1	7.791903(-7)
F1	10^{-8}	1419	109	0	1.221122(-6)
F2		1552	140	1	3.026956(-7)
F1	10^{-10}	3434	264	0	5.590682(-8)
F2		4588	416	1	1.803782(-8)

Table 6: Numerical Results for Problem 6