

OSFESOR Code – The Delay Differential Equation Tool “Improving Delay Differential Equations Solver”

Reyadh S. Naoum
Dept. Mathematics
College of Science
Baghdad University

Abbas Y. Al-Bayati Ann J. Al-Sawoor
Dept. of Mathematics
College of Computer Sciences and Mathematics
Mosul University

Received on: 15/12/2003

Accepted on: 22/3/2004

المخلص

بعد أن تم استعراض البرنامج RETAERD ، الذي قدمه العالمان هايرر و ورنر في عام 1995 بهدف حل المعادلات التفاضلية المتأخرة عددياً (DDEs) ، فلقد تم في هذا البحث استحداث البرنامج OSFESOR. تشير الكلمة OSFESOR إلى (حجم الخطوة المثلى في كل خطوة للبرنامج RETARD والذي يعد من التطبيقات الأوتوماتيكية للمعادلات التفاضلية المتأخرة بلغة Fortran 77 . وبناءً عليه فإنه، من خلال استخدام البرنامج الجديد أصبح من الممكن الآن حساب الحل لمعادلات تفاضلية متأخرة ، وذلك باستخدام حجم خطوة تكاملية مثلى ، H ، لكل خطوة لتحسين الدقة لأية نتيجة حاسوبية . باختصار إن استخدام برنامج OSFESOR يخدم في تحقيق أفضل النتائج الحاسوبية و تضمن للمستخدم ثقة الحسابات العلية.

ABSTRACT

After having reviewed the RETARD code, which was originally written by Hairer & Wanner in 1995 with the aim of solving delay differential equations (DDEs), a new arithmetic called OSFESOR code is presented in this paper. The OSFESOR (Optimal stepsize for each step of RETARD) code is a tool for automatic implementation of DDEs in Fortran 77. Consequently, by use of the OSFESOR code, it is possible during the run of a program to evaluate the solution of DDEs with optimal stepsize, H , for each step, to evaluate the accuracy of any result provided by the computer.

In short, the use of OSFESOR code serves to validate the result provided by a computer, and to assure the user of the reliability of scientific computations.

1- Introduction

The simulation of phenomena represents virtually all of scientific computation. Yet just what is the simulation of phenomena? It is the development of a scientific code, which translates a mathematical model and employs numerical methods to resolve it. RETARD code which was written by Hairer & Wanner, which is modified in 1995, [1] was concerned with Runge- Kutta methods for a system of first order ordinary differential equations of the form:

$$y'(t) = f(t, y(t), y(t-t), \dots) \quad t \geq 0 \quad \dots\dots\dots(1)$$

subject to the initial condition $y(t) = f(t)$ for $t_{\min} \leq t \leq 0$. The basic strategy investigated, there involves the adaptation of Runge – Kutta method for the ordinary differential equation :

$$y'(t) = F(t, y(t)) \quad \dots\dots\dots(2)$$

With prescribed initial value, we assume that the reader is familiar with Runge– Kutta methods for (2), the concept of their order and internal stage –orders and their representation by corresponding tableaux of the form:

$$\begin{array}{c} C \quad A \\ b \end{array} = \begin{array}{ccccccc} c_1 & a_{11} & . & . & . & a_{1v} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ c_v & a_{v1} & . & . & . & a_{vv} \\ & b_1 & . & . & . & b_v \end{array}$$

With an example of the fifth order Dpmand &Prince Runge– Kutta [2]

0	0	0	0	0	0	0	0
$\frac{1}{5}$	$\frac{1}{5}$	0	0	0	0	0	0
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	0	0	0	0	0
$\frac{4}{5}$	$\frac{44}{45}$	$\frac{-56}{15}$	$\frac{32}{9}$	0	0	0	0
$\frac{8}{9}$	$\frac{19372}{6561}$	$\frac{-25360}{2187}$	$\frac{64448}{6561}$	$\frac{-212}{729}$	0	0	0
1	$\frac{9017}{3168}$	$\frac{-355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$\frac{-5103}{18656}$	0	0
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	0
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$\frac{-2187}{6784}$	$\frac{11}{84}$	0

This code which is based on an explicit Runge-Kutta method for (2) of order (4) 5 due to Dormand & Prince is adapted to the delay differential equations of (1).

Thus, a specific Runge-Kutta method for solving a delay differential equation involves

- 1- A choice of Runge-Kutta tableau.
- 2- A choice of approximation method for interplant [2].

2-Interpolation:

Runge–Kutta method can be adapted for DDEs, such that we consider the fifth-order Dormand & Prince explicit RK method. There are several possible choices of dense – output for RK method [3],[4]&[5] :

- 1- Continuous (embedded) extensions.
- 2- Natural continuous extensions (NCEs).
- 3- Low order Hermite interplant.
- 4- Highly continuous extensions (HCEs).

5- High order Hermite interplant.

6- Multi –step continuous extensions.

Some of these approximation schemes have the following continuous extensions : [5]

*** Third Order Hermite Formula:**

The continuous extension (the coefficients for the third-order Hermite interplant) is:

$$b_1(q) = \frac{157}{192}q^3 - \frac{221}{128}q^2 + q \quad ,$$

$$b_2(q) = 0 \quad ,$$

$$b_3(q) = \frac{-1000}{1113}q^3 + \frac{500}{371}q^2 \quad ,$$

$$b_4(q) = \frac{-125}{96}q^3 + \frac{125}{64}q^2 \quad ,$$

$$b_5(q) = \frac{2187}{3392}q^3 - \frac{6561}{6784}q^2 \quad ,$$

$$b_6(q) = -\frac{11}{42}q^3 + \frac{11}{28}q^2 \quad ,$$

$$b_7(q) = q^3 - q^2 \quad .$$

*** Fourth Order Continuous Embedded Formula:**

$$b_1(q) = \frac{-1163}{1152}q^4 + \frac{1039}{360}q^3 - \frac{1337}{480}q^2 + q \quad ,$$

$$b_2(q) = 0 \quad ,$$

$$b_3(q) = \frac{7580}{3339}q^4 - \frac{18728}{3339}q^3 + \frac{4216}{1113}q^2 \quad ,$$

$$b_4(q) = \frac{-415}{192}q^4 + \frac{9}{2}q^3 - \frac{27}{16}q^2 \quad ,$$

$$b_5(q) = \frac{-8991}{6784}q^4 + \frac{2673}{2120}q^3 - \frac{2187}{8480}q^2 \quad ,$$

$$b_6(q) = \frac{187}{84}q^4 - \frac{319}{105}q^3 + \frac{33}{35}q^2 \quad ,$$

$$b_7(q) = 0 \quad .$$

*** Fifth Order Hermite Formula:**

$$b_1(q) = \frac{29}{16}q^5 - \frac{81685}{14208}q^4 + \frac{24433}{3552}q^3 - \frac{6839}{1776}q^2 + q \quad ,$$

$$b_2(q) = 0 \quad ,$$

$$b_3(q) = -\frac{4000}{371}q^5 + \frac{1245700}{41181}q^4 - \frac{398800}{13727}q^3 + \frac{413200}{41181}q^2 ,$$

$$b_4(q) = -\frac{125}{8}q^5 + \frac{83775}{2368}q^4 - \frac{44725}{1776}q^3 + \frac{225}{37}q^2 ,$$

$$b_5(q) = \frac{6561}{848}q^5 - \frac{4428675}{251008}q^4 + \frac{798255}{62752}q^3 - \frac{98415}{31376}q^2 ,$$

$$b_6(q) = -\frac{22}{7}q^5 + \frac{527571}{73556}q^4 - \frac{285659}{55167}q^3 + \frac{23529}{18389}q^2 ,$$

$$b_7(q) = 4q^5 - \frac{21872}{2627}q^4 + \frac{14847}{2627}q^3 - \frac{3483}{2627}q^2 ,$$

$$b_8(q) = \frac{-40}{37}q^4 + \frac{80}{37}q^3 - \frac{40}{37}q^2 ,$$

$$b_9(q) = 16q^5 - 40q^4 + 32q^3 - 8q^2 .$$

A Runge–Kutta method can be combined with a continuous extension to produce a continuous Runge - Kutta method (CRK). The continuous extension of a Runge – Kutta process is found as one of the standard techniques for obtaining dense-output in the solution of (2). It is based upon the continuous Runge-Kutta (CRK) triple $(c, A, b(q))$ featured, with an example, in the tableau in [6]

C A

$\theta \quad b(\theta)$

Where we have $A=[A_{ij}] \in R^{m \times m}$, $b(q)=[b_1(q), \mathbf{K}, b_m(q)]^T$ and $C=[C_1, \mathbf{K}, C_m]^T \in R^m$ for example,

0	0	0	0	0	0	0	0
$\frac{1}{5}$	$\frac{1}{5}$	0	0	0	0	0	0
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$	0	0	0	0	0
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$	0	0	0	0
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$	0	0	0
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	0	0
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$b_1(q)$	$b_2(q)$	$b_3(q)$	$b_4(q)$	$b_5(q)$	$b_6(q)$	$b_7(q)$

where $b_1(q), b_2(q), b_3(q), b_4(q), b_5(q), b_6(q)$ and $b_7(q)$ are the fourth order continuous embedded formula .

The RK parameter are “formally explicit” if $a_{ij} = 0$ for $j \geq i$ and will be called “local” if $C_i \in [0,1]$ for all i.

A CRK triple allows one to obtain a formulate (dense - output) for the numerical solution of a DDEs.

Now after having reviewed the RETARD code, which solving a system of first order DDEs(1), a new arithmetic called OSFESOR code is presented in this paper .

3- scientific computation of the optimal stepsize for each step in OSFESOR code :

Consider the method error e_m and the error due to the propagation of round-off error, called the computation error e_c here. It is well known that when the discretizing step h decreases, e_m decreases and that, on the contrary, when h increases, e_m also increases. It has been shown that, when h increases, e_c decreases, and that when h decreases, e_c increases. This means that e_m & e_c act in the opposite way . Thus the best approximation of the solution that can be obtained on a computer corresponds to an optimal discretizing step h .

The scientific computation of the optimal stepsize h for each step in OSFESOR code can be computed by the following outlines:

3.1-The Outlines:

The computation of optimal stepsize h requires three phases:

- (i) The estimation of round-off error e_c ;
- (ii) The evaluation of the truncation error e_m ;
- (iii) The computation of the optimal stepsize h .

* The estimation of round-off error e_c is consequently computed by the equation:

$$e_c = 10^{-(n+1)}, \text{ Where } n \text{ is an integer bounded from below.}$$

* The estimation of the truncation error e_m is obtained by using Euler's method where the e_m for this method is given by the equation:

$$e_m = 2|v_1 - v_2|$$

v_1 is the value of $y(x_k+h_k)$ integrated over the interval $[x_k, x_k+h_k]$ with stepsize h_k ,

v_2 is the value of $y(x_k+h_k)$ integrated over the interval $[x_k, x_k+h_k]$ with stepsize $\frac{1}{2}h_k$.

*The optimal stepsize h is evaluated for each interval $[x_k, x_k+h_k]$ as follows

- (i) $h_k = 1.5h_{k-1}$
- (ii) if $(e_m \leq e_c)$ or $(e_m = 0)$ then $e_m = e_c$
- (iii) if $(e_m = e_c)$ then $h_k = h_{k-1}/\Omega$ where $0 < \Omega \leq 5$
- (iv) if $(e_c < e_m)$ then $h_k = 4h_{k-1}/3.0001$

4-Driver for RETARD & OSFESOR:

Example (1):

A Kermack-Mckendrick model of an infectious disease with periodic outbreak is discussed in ([9],[10]&[11])

$$y_1'(t) = -y_1(t)y_2(t-1) + y_2(t-10) \quad t \geq 0$$

$$y_2'(t) = y_1(t)y_2(t-1) - y_2(t) \quad t \geq 0$$

$$y_3'(t) = y_2(t) - y_2(t-10) \quad t \geq 0$$

On $[0,40]$ with history

$$y_1(t) = 5 \quad , \quad y_2(t) = 0.1 \quad , \quad y_3(t) = 1 \quad \text{for } t \leq 0.$$

Analytical solution: not available

Sources: E.Hairer , G.Wanner &Norsett,solving ODEs
1,sipringer series in comp. Math.vol.8(1980).(see[9])

Other information: This system of DDEs describes a disease
model a reference solution is (see[9])

$$y_1(40)=0.0912491205663460$$

$$y_2(40)=0.0202995003350707$$

$$y_3(40)=5.98845137909849$$

The results in the table below obtained with the use of
RETARD & OSFESOR codes:

The result of RETARD code:

x	$y_1(x)$
0.00	0.5000000000E+01
15.29	0.4539376456E+01
25.22	0.2524799457E+00
35.10	0.3610797907E+00
Xend	0.91255445550E-01

xend	$y_1(x)$	$y_2(x)$	$y_3(x)$
xend	0.9125544555E-01	0.2029882456E-01	0.5988445730E+01
Error at xend	0.000006324983654	0.0006887452649293	0.000005649098449

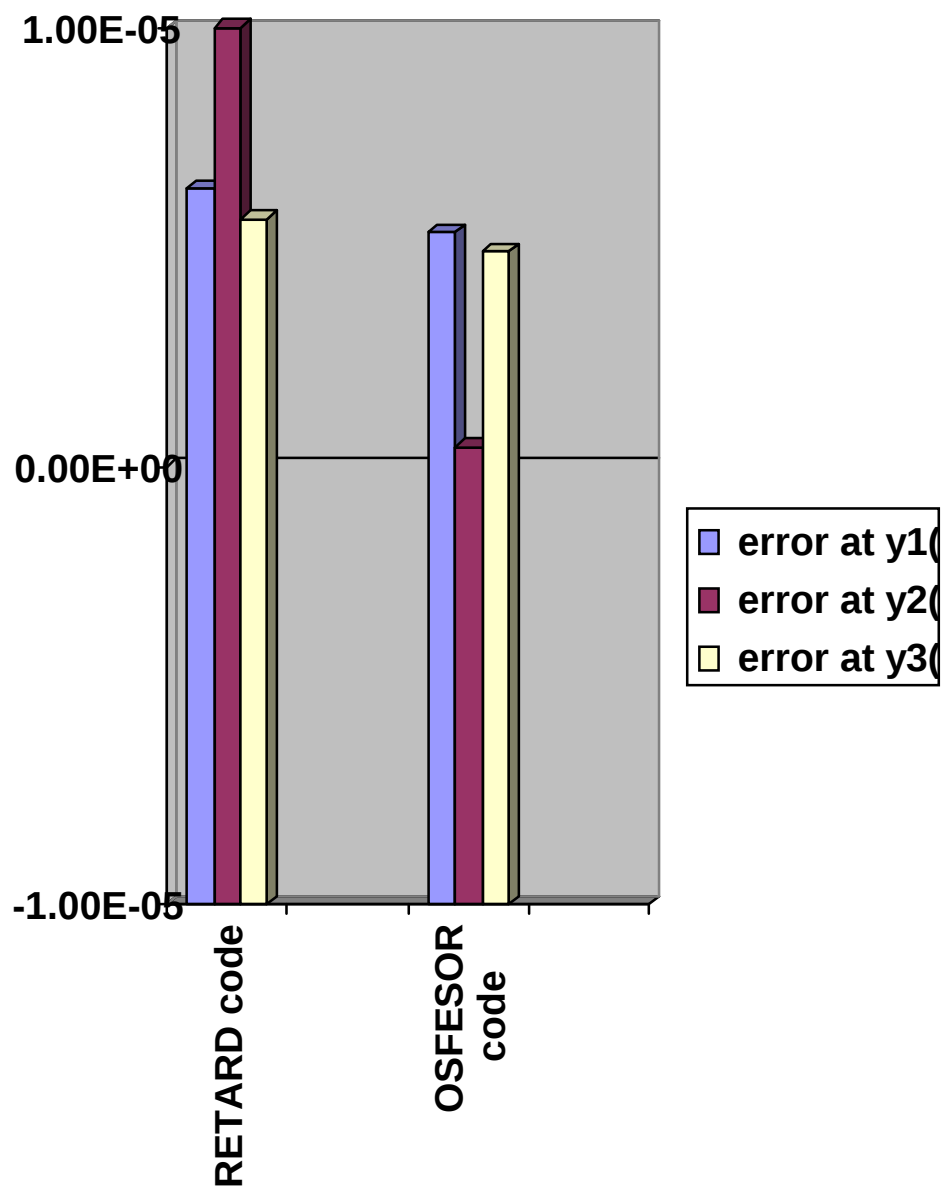
Tol=.10D-04	Fcn=586	Step=97	Accpt=89	Rejct=8
-------------	---------	---------	----------	---------

The result of OSFESOR code:

x	$y_1(x)$
0.00	0.5000000000E+01
15.29	0.4539344282E+01
25.22	0.2524774858E+00
35.10	0.3612313767E+00
xend	0.9125445368E-01

xend	$y_1(x)$	$y_2(x)$	$y_3(x)$
xend	0.9125445368E-01	0.2029905626E-01	0.5988446490E+01
Error at xend	0.000005333113654	0.0000004440750707	0.00000488909849

Tol=.10D-04	Fcn=584	Step=97	Accpt=89	Rejct=8
-------------	---------	---------	----------	---------



Example (2):

A chronic granulocytic leukemia model is discussed in [9]

$$y_1'(t) = \frac{1.1}{1 + \sqrt{10}(y_1(t-20))^{5/4}} - \frac{10y_1(t)}{1 + 40y_2(t)} \quad t \geq 0$$

$$y_2'(t) = \frac{100y_1(t)}{1 + 40y_2(t)} - 2.43y_2(t) \quad t \geq 0$$

on [0,100] with history

$$y_1(t) = 1.057670273, \quad y_2(t) = 1.0307134913 \quad \text{for } t \leq 0.$$

Analytical solution: not available

Sources: H.J.Oberle & H.J.Pesch , Numerical Treatment of Delay Diff. Eqns. By Hermite Interpolation, Numer. Math. Vol.37 (1981), pp.235-255.

Other information: This system of DDEs describes a model of chronic granulocytic leukemia. A reference solution is (see[9])

$$y_1(100) = 0.0876801107411822$$

$$y_2(100) = 0.2937685943262440$$

The results in the table below obtained with the use of RETARD & OSFESOR codes:

The result of RETARD code:

x	y ₁ (x)
0.00	0.3525567452E+00
20.1	0.8230105418E+01
30.17	0.3934964051E+01
40.09	0.1203972631E+01
50.40	0.3446209588E+00
60.23	0.5644294368E+00
80.59	0.2117511652E+01
xend	0.8602397961E+00

OSFESOR Code – The Delay Differential...

xend	y₁(x)	y₂(x)
xend	0.8602397961E+00	0.9334859861E+00
Error at xend	0.7725596853588178	0.639717391773756

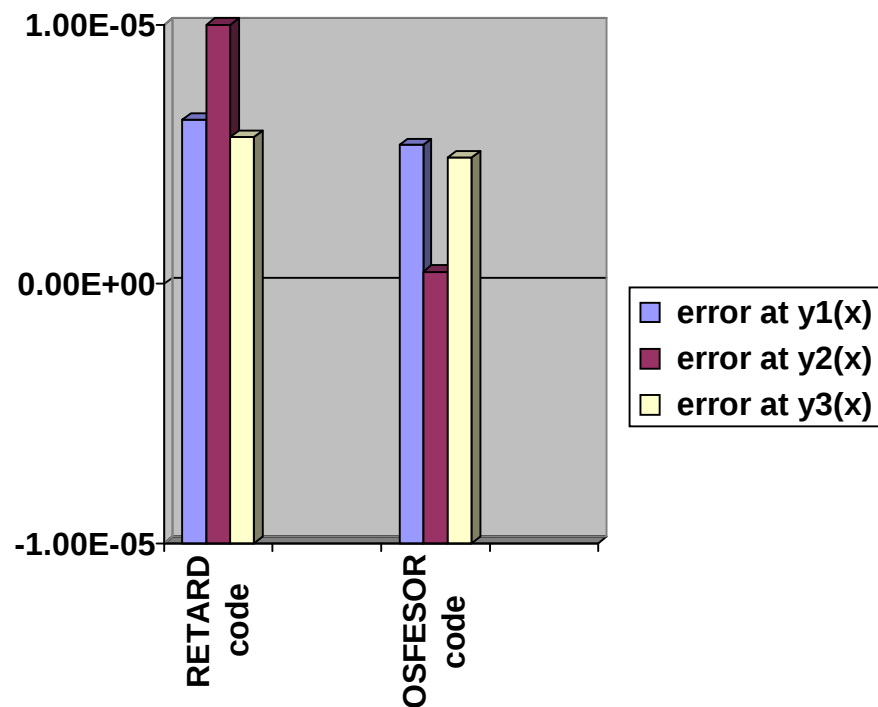
Tol=.10D-04	Fcn= 1382	Step= 230	Accpt= 226	Rejct= 4
-------------	-----------	-----------	------------	----------

The result of OSFESOR code:

x	y₁(x)
0.00	0.3525567452E+00
20.1	0.8230059200E+01
30.17	0.3934929559E+01
40.09	0.1204029277E+01
50.40	0.3446322032E+00
60.23	0.5643704969E+00
80.59	0.2117545664E+01
xend	0.8602337744E+00

xend	y₁(x)	y₂(x)
xend	0.8602337744E+00	0.9334825907E+00
Error at xend	0.7725536636588178	0.639713996373756

Tol=.10D-04	Fcn= 1382	Step= 230	Accpt= 226	Rejct= 4
-------------	-----------	-----------	------------	----------



5- The OSFESOR code:

```

      SUBROUTINE OSFESOR(N,FCN,X,Y,XEND,
&          RTOL,ATOL,ITOL,
&          SOLOUT,IOUT,
&          WORK,LWORK,IWORK,LIWORK,LRCONT,LICONT,
&          RPAR,IPAR,IDID)
C  NUMERICAL SOLUTION OF A SYSTEM OF FIRST ORDER DELAY
C  ORDINARY DIFFERENTIAL EQUATIONS  $Y'(X)=F(X,Y(X),Y(X-A),...)$ .
C  THIS IMPROVED CODE IS BASED ON AN EXPLICIT RUNGE-
C  KUTTA METHOD OF ORDER (4)5 DUE TO DORMAND & PRINCE
C  (WITH OPTIMAL STEPSIZE CONTROL c FOR EACH STEP AND
C  DENSE OUTPUT).

```

6-Conclusions:

The main advantage of using OSFESOR code is to develop DDEs methods making it possible to estimate the optimal step or mesh at each computation step, so that the best possible informational solution can be obtained. Thus we feel that the one can now use tools, such as the OSFESOR code, for validating the results provided by the computer and should also forcefully demand that manufactures creating arithmetic units and compilers include tools for numerical validation.

REFERENCES

- [1] E.Hairer, S.P.Norsett and G.Wanner.solving ordinary differential eqns. I. Nonstiff problems .2nd edition.springer series in computational Mathematics, springer-verlag (1993), version of September 30,1995.
- [2] C.A.H.Paul & C.T.H.Baker. Stability boundaries revisited -Runge- Kutta methods for delay differential equations, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.205 (revised), December (1991).
- [3] C.T.H.Baker, C.A.H.Paul & D.R.Wille'. Issues in the numerical Solution of evolutionary delay differential equations, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.248, April (1994).
- [4] C.A.H.Paul. Developing a delay differential equation solver, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.204, December (1991).
- [5] C.A.H.Paul & C.T.H.Baker.explicit Runge -Kutta methods for the numerical solution of singular delay differential equations, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.212 (revised), April (1992).
- [6] C.T.H.Baker. Retarded differential equations, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.365, Oct (2000).
- [7] J.Vignes. A stochastic approach to the analysis of round-off error propagation-a survey of the CESTAC method centre international de rencontres mathe'm atiques,

- Marseilles France. 9-10-11 April -Avril (1996), pp.: 233-251.
- [8] C.T.H.Baker, C.A.H.Paul & D.R.Wille. A Bibliography on the numerical solution of delay differential equations, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.269, version 1.0, June (1995).
 - [9] C.A.H.Paul.A Test of functional differential equations, university of Manchester /UMIST Manchester centre for computational mathematics, numerical analysis reports, report no.243, February (1994).
 - [10] L.F.Shampine &S.Thompson.Solving DDEs in MATLAB, applied numer. Math.,37(4):441-458,(2001).
 - [11] L.F.Shampine &S.Thompson. Solving delay differential equations with dde23, southern Methodist University, March 23, (2000).