The evaluation of numerical software for delay differential equations

W.H. Enright and H. Hayashi
University of Toronto
Department of Computer Science, Toronto M5S 3G4, Canada.
Telephone: +1-416-978-[2980,3965]. Fax: +1-416-978-1931.
Email: {enright,hiroshi}@cs.toronto.edu

Abstract

We have been involved in the development and evaluation of software for initial value problems in ordinary differential equations for several years. Experience gained in our early testing and comparisons of nonstiff and stiff methods provided us with key insights and motivation for the design and development of two software packages, NSDTST and STDTST, that can be used for the evaluation of numerical methods for initial value problems. These packages have been widely distributed and have proved to be particularly valuable when assessing the relative advantages of various strategies or heuristics involved in implementing a new method.

Recently we have developed a new approach for the numerical solution of delay differential equations (DDEs) based on the use of continuous Runge-Kutta formulas. In implementing our approach as a general purpose numerical method we recognized the need for software tools, similar to those provided by the NSDTST package, that could be used in the evaluation of numerical methods for DDEs. In this paper we will discuss how we adapted and extended the NSDTST package so it could be used for this purpose. We will present examples of how the resulting test package, DDETST, can be used to assess different implementations of our approach for solving DDEs as well as other existing general purpose methods.

Keywords

Software evaluation, delay differential equations, neutral equations, defect control

1 INTRODUCTION

For over two decades, at the University of Toronto we have been interested in the development of effective numerical software for initial value problems in ordinary differential equations. An important component of this ongoing activity has been the development of

R. F. Boisvert (ed.), *Quality of Numerical Software*

tools that can be used to assess the performance of an initial value method on a particular class of test problems. Two testing packages, NSDTST and STDTST (for assessing methods suitable for nonstiff problems and stiff problems respectively), are described in Enright and Pryce (1987). In developing these assessment tools we attempted, as much as possible, to ensure that the package's monitoring of a problem's solution does not itself inhibit (or affect) the performance. A consequence of this is that one must be careful when comparing different methods on the basis of the assessments determined by the testing packages (as no attempt is made to ensure that methods are 'doing the same thing'). To address this inherent difficulty, the packages provide, as an option, the ability to produce 'normalized efficiency statistics' which are more appropriate for use when comparing the performance of different methods.

Recently we have become interested in the analysis and development of methods for delay differential equations (DDEs). Advances in the area of continuous Runge-Kutta methods for initial value problems (see for example Enright, Jackson, Nørsett and Thomsen (1986), Dormand and Prince (1986) and Shampine (1985)) have provided a natural approach that can be followed to develop effective methods for DDEs. In carrying out the implementation component of this research we have developed a testing package, based on the structure and design of NSDTST, that can be used for the evaluation and assessment of numerical methods for DDEs. In this paper we will discuss the design decisions and implementation issues that we addressed as well as give examples of how the final package DDETST can be effectively used.

The first issue that one must address when considering the assessment of methods for DDEs is the identification and classification of suitable problems that a 'general purpose method' can be expected to solve. In the case of initial value problems in ODEs it is generally accepted that there are two classes of general purpose methods—those suitable for nonstiff problems and those suitable for stiff problems. For DDEs there is no such consensus. Delay problems can be partitioned into categories depending on a number of different factors and the relative effectiveness of a particular method can be very sensitive to the partitioning. We will consider systems of equations with the possibility of multiple delays in the solution or derivative values. That is, problems of the form

$$\begin{aligned} y' &= f(t, y(t), y(t-\sigma_1), \cdots, y(t-\sigma_k), \\ &\qquad y'(t-\sigma_{k+1}), \cdots, y'(t-\sigma_{k+\ell})), \text{ for } t_0 \le t \le t_F, \\ y(t) &= \phi(t), \;\; y'(t) = \phi'(t), \; t \le t_0, \end{aligned} \tag{1}$$

where $\sigma_i \equiv \sigma_i(t, y(t)) \ge 0$ for $i = 1, 2, \cdots, k+\ell$ and $y(t) \in R^n$. When $\ell = 0$ the problem is considered a retarded differential equation (RDE) and when $\ell > 0$ it is a neutral differential equation (NDE).

A general purpose method for solving problems of this form must, in order to sample the differential equation, be capable of approximating the true solution $y(t)$ and the derivative $y'(t)$ at any point $t \in (t_0, t_F)$. That is, there must be an underlying continuous differentiable approximation to $y(t)$ associated with the numerical method. If $z(t)$ is this underlying continuous approximation to $y(t)$ then $z(t)$ should 'almost satisfy' (1) and we can associate a defect, $\delta(t)$, with $z(t)$ by

$$\delta(t) = z'(t) - f(t, z(t), z(t-\sigma_1), \cdots, z(t-\sigma_k), \; z'(t-\sigma_{k+1}), \cdots, z'(t-\sigma_{k+\ell})). \tag{2}$$

Our preliminary testing of existing methods for DDEs indicated that the effectiveness (or in some cases even the applicability) of a particular method could be very sensitive to three factors:

- the existence of derivative delays ($\ell > 0$),
- the existence of state dependent delays ($\sigma_i(t, y(t))$ depends on $y(t)$ for some i),
- the existence of small or vanishing delays (either $\sigma_i(t^*, y(t^*)) = 0$ for some $t^* \in (t_0, t_F)$ or $|\sigma_i(t, y(t))|$ becomes small relative to the expected stepsize).

With this in mind we partitioned our candidate test problems according to these factors. We gathered test problems from the literature relying particularly on test problems used by Neves and Thompson (1992b), and Paul (1994). In some cases where the problem class contained only a single problem we introduced new test problems of our own. We now have a collection of twenty-three test problems grouped in eight problem classes:

- Class A: RDEs with time-dependent delay,
- Class B: RDEs with small time-dependent delay,
- Class C: RDEs with state-dependent delay,
- Class D: RDEs with small state-dependent delay,
- Class E: NDEs with time-dependent delay,
- Class F: NDEs with small time-dependent delay,
- Class G: NDEs with state-dependent delay,
- Class H: NDEs with small state-dependent delay.

The complete specification of the test problems for each of these classes is given in the Appendix.

For about one half of our test problems an analytic closed form expression for the true solution is known. In the remaining cases we were able to determine, using a reliable numerical method (implemented in quadruple precision with an error tolerance of 10^{-12}) a continuous approximation to the true solution that we feel confident is accurate to more than ten decimal digits. (It is interesting to note that the associated piecewise polynomial (continuous extension) is of degree six on each subinterval and the maximum number of subintervals required to provide this accuracy was 9600. This indicates the importance of the use of a high order method for computing the reference solution.)

2 THE DDETST PACKAGE

The DDETST package shares the same design and overview as the original NSDTST package. The package monitors each step in the solution of a test problem at a prescribed error tolerance and records and reports the following statistics:

- the total computer time required to solve the problem,
- the number of derivative evaluations required to solve the problem,
- the number of successful steps required to solve the problem,
- the endpoint global error,

- the maximum global error observed over the interval of integration,
- the maximum defect observed over the interval of integration,
- the fraction of steps for which the maximum defect exceeds the tolerance.

Note that the first four statistics are always reported while the others are optional. The maximum global error and the maximum defect are defined in terms of the underlying continuous approximation, $z(t)$, and are based on a number of sampled values per time step (the actual number of sampled values used is a parameter of the testing package which must be set by the user).

After the completion of each problem over a range of error tolerances statistics quantifying the 'smoothness' (or robustness) of the method are determined (if requested). For this purpose the method is assessed on how well it has been able to satisfy:

$$\text{error} = C \cdot TOL^{E}, \tag{3}$$

for some $C > 0$ and E 'close' to one, over the range of tolerances that were specified. These smoothness statistics are justified and discussed in detail in Enright and Pryce (1987) for initial value problems and they form the basis for the definition of the 'normalized efficiency statistics' which are suitable for use when comparing different methods. Note that smoothness statistics based on (3) are optional and are available for three measures of error (endpoint global error, maximum global error and maximum defect).

After the completion of each problem class and after the completion of all problems corresponding summary statistics are reported. These summaries can be useful but must be interpreted carefully as they can be distorted by failure or anomalous behavior on one problem.

The overall control structure and control interfaces of the DDETST package is represented in Figure 1 where the key subroutines that comprise DDETST are indicated by their names (in upper case) and the routines that must be supplied by the user are in lower case enclosed in rectangles. The routines of the testing package and their principal tasks are:

DDTST	–	organizes and monitors the overall collection of statistics,
CNTROL	–	supervises the solution of a single integration,
STATS	–	monitors the code being tested and passes statistics via COMMON to DDTST and CNTROL,
YSOL	–	computes the 'true' solution for any t in the range of integration,
IVALU	–	specifies the number of equations, integration interval, the number of delay arguments, and weights for scaling,
EVALU	–	specifies the 'true' solution at the endpoint,
DVALU	–	evaluates the delay arguments $t - \sigma_i(t, y(t))$ for $i = 1, 2, \cdots, k + \ell$,
IFCN, DIFCN	–	evaluates the initial functions $\phi(t)$ and $\phi'(t)$ for $t \le t_0$,
FCN	–	evaluates the differential equation.

In the main program (labeled 'user's program' in Figure 1), the particular problems, range of tolerances and level of detailed statistics requested are selected. For example,

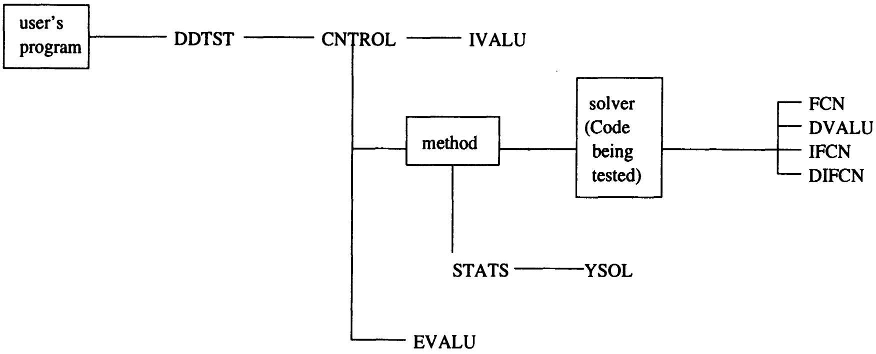

Figure 1 Overall control structure of DDETST.

```
C       .. Local Scalars ..
        REAL            FLAG
        INTEGER         IOUT
        CHARACTER*80    TITLE
C       .. Local Arrays ..
        REAL            TOL(11)
        INTEGER         IDLIST(80), OPTION(10)
C       .. External Functions ..
        REAL            CONST
        EXTERNAL        CONST
C       .. External Subroutines ..
        EXTERNAL        DDTST
C       .. Data statements ..
        DATA            OPTION/5, 0, 1, 0, 100, 100, 4*0/
        DATA            TOL/1.E-4,1.E-6,1.E-8,1.E-10,7*0.E0/
        DATA            IDLIST/11,12,0,21,22,75*0/

C       .. Executable Statements ..
C     SAMPLE DRIVER FOR DDTST, WITH TWO GROUPS CONSISTING OF
C     PROBLEM CLASSES A,B SOLVED IN SCALED FORM,
C     AT FOUR TOLERANCES, WITH OPT=5 AND NORMEF=0.

        TITLE = 'DDVERK'
        IOUT = CONST(3)
        OPEN(IOUT, FILE="DDETEST-DDVERK")
        CALL DDTST(TITLE,OPTION,TOL,IDLIST,FLAG)
        CLOSE(IOUT)

        END
```

Figure 2 Sample user's program.

- Declare Workspace required by SOLVER
- Set options of SOLVER (in particular set one-step mode)
- for each successful step do
  - invoke SOLVER ( )
  ... - if SOLVER in trouble then exit
  - if max global err requested then determine the approximation z at kg poi
  - if max defect requested then measure $\delta(t)$ at kd points
  - invoke STATS ( )

  end do

Figure 3 Overview of METHOD routine.

Figure 2 presents the 'user's program' for a run assessing the code DDVERK on two problem classes, over four tolerances, with detailed assessment of the maximum observed global error and maximum observed defect (indicated by setting IOPT(1) = 5).

The method being tested is usually presented as two subroutines: METHOD and SOLVER. METHOD is a special driver supplied by the user with a standard calling sequence dictated by the testing package. Its purpose is to set up the workspace and options of the underlying method, encoded in the routine SOLVER, so that it will return after every successful step. A generic overview of METHOD is presented in Figure 3.

Note that the routines defining the twenty-three test problems and the associated underlying true solution comprise a useful package on their own and can be called directly by users.

3 EXAMPLES OF USE OF DDETST

In Enright and Hayashi (1996a; 1996b) we have developed and analyzed a method for DDEs based on the use of continuous explicit Runge-Kutta formulas with defect-based error and stepsize control. The method DDVERK estimates the maximum magnitude of the defect associated with the numerical solution $z(t)$ on each step of the integration and attempts to ensure that this is bounded by the prescribed error tolerance. That is, on each successful step

$$E \leq TOL ,$$

where E is an estimate of the maximum defect associated with that step and is therefore assumed to satisfy

$$E \cong \max_{t \leq s \leq t+h} \{\| \delta(s) \|\} .$$

The stepsize changing strategy and the discontinuity-locating strategy, two of the most critical components of DDE solver, are based on the reliability of this estimate. In deciding on what formula to use to determine E one must inevitably consider a trade-off between cost and reliability. (For a detailed investigation of the equivalent trade-offs in initial value methods see Enright (1993).) In developing DDVERK we considered three possible definitions for E: one based on one sample point per step,

$$E1 = \| \delta(t + t_1 h) \|$$

for a carefully chosen value of t_1; one based on the use of two sample points per step,

$$E2 = \max\{\| \delta(t + \bar{t}_1 h) \|, \| \delta(t + \bar{t}_2 h) \|\}$$

where $\bar{t}_1$ and $\bar{t}_2$ are carefully chosen; and one based on the use of a new, more accurate continuous approximation, $\hat{z}(t)$, requiring three extra derivative evaluations per step but yielding an asymptotically ($h \to 0$) correct estimate

$$E3 = \| \hat{\delta}(t + t^* h) \| \; .$$

Table 1 presents a summary of the statistics generated by DDETST for three implementations of DDVERK corresponding to the use of these three estimates on problem A1. Note that the extra cost per step but improved reliability of E2 and E3 is clearly reflected in the statistics. Figure 4 gives a graphical representation of the same information. (Note that the solution to this problem approaches a periodic steady state with the specified length of integration spanning several periods. In this situation the long-term accuracy of the numerical solution can be very sensitive to the errors introduced in the initial interval where low order discontinuities can have an effect. The rather large values for the observed maximum global error, reported in Table 1, are due to this sensitivity.) Table 2 and Figure 5 present the corresponding overall summary statistics for the three implementations on nineteen of the twenty-three test problems. (Problems C2, E1, F1, and F2 are excluded because at least one of the three implementations failed to solve these problems at tolerance of 10^{-10}.)

Interface routines for methods developed elsewhere, such as ARCHI (Paul, 1995), DRKLAG5 (Neves and Thompson, 1992a), and DRKLAG6 (Corwin and Thompson, 1996) have been written and detailed assessments carried out using DDETST. It is too early to report any detailed comparisons but weakness of particular methods have been identified. For example, some methods cannot handle neutral problems with vanishing delays effectively. We are expecting that distribution of DDETST will lead to the development and release of improved versions of existing methods.

REFERENCES

Castleton, R. N. and Grimm, L. J. (1973). A first order method for differential equations of neutral type, *Math. Comput.* **27**, 571–577.

Corwin, S. C. and Thompson, S. (1996). DKLAG6: Solution of systems of functional differential equations with state dependent delays, *Technical Report 96-002*, Computer Science Department, Radford University, Radford.

Dormand, J. R. and Prince, P. J. (1986). Runge-Kutta triples, *Computers Math. Applic.* **12A**(9), 1007–1017.

El'sgol'ts, L. E. and Norkin, S. B. (1973). *Introduction to the Theory and Application of Differential Equations with Deviating Arguments*, Academic Press, New York.

Table 1 Statistics reported for DDVERK using different defect estimates on A1

	TOL	FCN CALLS	STEPS	MAX GLB ERR	MAX DEF	FRACT DEF > TOL
E1 – DDVERK	10^{-2}	573	52	1.9	0.48	0.00
	10^{-4}	1345	79	169.7	9.32	0.38
	10^{-6}	2524	164	24.6	8.61	0.38
	10^{-8}	5243	346	15.3	11.57	0.53
	10^{-10}	9688	718	12.9	18.46	0.63
E2 – DDVERK	10^{-2}	625	52	1.9	0.48	0.00
	10^{-4}	1875	94	414.4	1.73	0.05
	10^{-6}	3300	191	54.6	2.70	0.11
	10^{-8}	6288	402	11.9	1.53	0.09
	10^{-10}	11 851	837	12.3	1.55	0.13
E3 – DDVERK	10^{-2}	729	52	1.4	0.47	0.00
	10^{-4}	2172	91	407.8	3.56	0.11
	10^{-6}	3966	191	184.4	9.02	0.07
	10^{-8}	7685	410	30.8	1.62	0.06
	10^{-10}	14 893	860	6.4	1.76	0.06

Table 2 Overall summary Statistics reported for DDVERK using different defect estimates

	TOL	FCN CALLS	STEPS	MAX DEF	FRACT DEF > TOL
E1 – DDVERK	10^{-2}	10 440	686	19.30	0.03
	10^{-4}	15 427	1058	9.32	0.05
	10^{-6}	25 680	1861	8.61	0.06
	10^{-8}	47 949	3615	12.75	0.09
	10^{-10}	90 054	7279	18.46	0.12
E2 – DDVERK	10^{-2}	11 292	696	1.78	0.02
	10^{-4}	17 188	1083	2.09	0.02
	10^{-6}	29 046	1934	2.70	0.02
	10^{-8}	53 340	3740	1.53	0.02
	10^{-10}	98 524	7453	3.14	0.02
E3 – DDVERK	10^{-2}	12 607	699	2.13	0.06
	10^{-4}	19 632	1101	3.56	0.03
	10^{-6}	33 181	1964	9.02	0.02
	10^{-8}	62 416	3842	2.41	0.02
	10^{-10}	121 217	7816	3.20	0.02

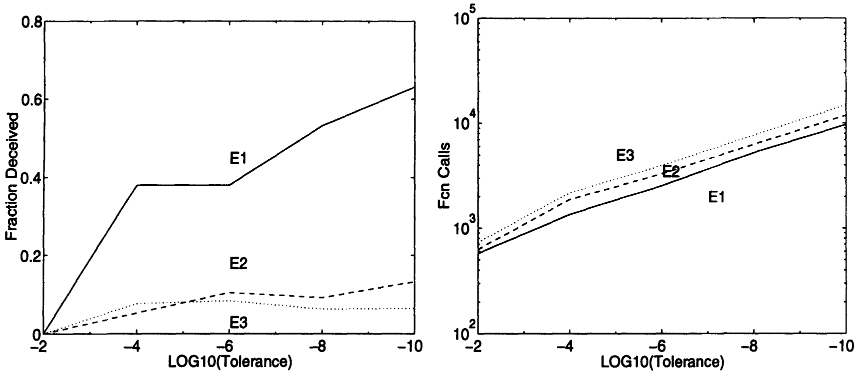

Figure 4 Problem: A1.

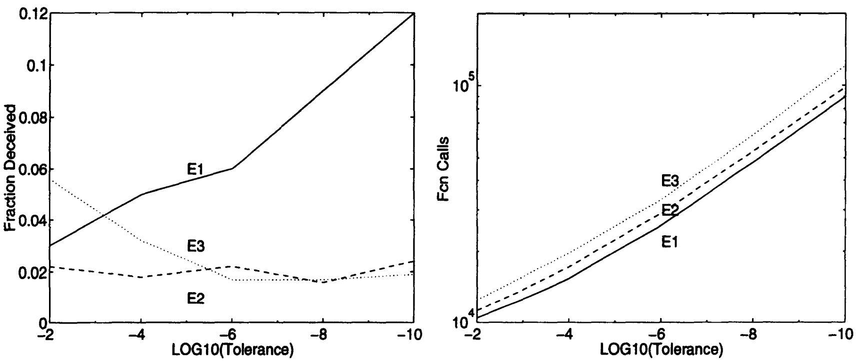

Figure 5 Overall summary statistics.

Enright, W. H. (1993). The relative efficiency of alternative defect control schemes for high order continuous Runge-Kutta formulas, *SIAM J. Numer. Anal.* **30**(5), 1419–1445.

Enright, W. H. and Hayashi, H. (1996a). Convergence analysis of the solution of retarded and neutral delay differential equations by continuous numerical methods, *Submitted to SIAM J. Numer. Anal.* .

Enright, W. H. and Hayashi, H. (1996b). A delay differential equation solver by a continuous Runge-Kutta method, *In preparation* .

Enright, W. H., Jackson, K. R., Nørsett, S. P. and Thomsen, P. G. (1986). Interpolants for Runge-Kutta formulas, *ACM Trans. Math. Soft.* **12**, 193–218.

Enright, W. H. and Pryce, J. (1987). Two FORTRAN packages for assessing initial value methods, *ACM Trans. Math. Soft.* **13**(1), 1–27.

Gatica, J. and Waltman, P. (1982). A threshold model of antigen antibody dynamics with fading memory, *in* Lakshmikantham (ed.), *Nonlinear phenomena in mathematical science*, Academic Press, New York, pp. 425–439.

Hayashi, H. (1996). *Numerical solution of retarded and neutral delay differential equations using continuous Runge-Kutta methods*, PhD thesis, Department of Computer Science,

University of Toronto, Toronto, Canada.
Jackiewicz, Z. (1981). One step methods for the numerical solution of Volterra functional differential equations of neutral type, *Applicable Anal.* **12**, 1–11.
Kuang, Y. (1991). On neutral delay logistics Gauss-type predator-prey systems, *Dyn. Stab. Systems* **6**, 173–189.
Kuang, Y. and Feldstein, A. (1991). Boundedness of solutions of a nonlinear nonautonomous neutral delay equation, *J. Math. Anal. Appl.* **156**, 293–304.
Mackey, M. C. and Glass, L. (1977). Oscillation and chaos in physiological control systems, *Science* **197**, 287–289.
Mahaffy, J. M., Bélair, J. and Mackey, M. C. (1996). Hematopoietic model with moving boundary condition and state dependent delay, *Private communication* .
Neves, K. W. (1975). Automatic integration of functional differential equations: An approach, *ACM Trans. Math. Soft.* **1**, 357–368.
Neves, K. W. and Thompson, S. (1992a). DKLAG5: Solution of systems of functional differential equations with state dependent delays, *Technical Report 92-003*, Computer Science Department, Radford University, Radford.
Neves, K. W. and Thompson, S. (1992b). Solution of systems of functional differential equations with state dependent delays, *Technical Report TR-92-003*, Computer Science, Radford University.
Paul, C. A. H. (1994). A test set of functional differential equations, *Technical Report 243*, The Department of Mathematics, The University of Manchester, Manchester, England.
Paul, C. A. H. (1995). A user-guide to Archi: An explicit Runge-Kutta code for solving delay and neutral differential equations, *Technical Report 283*, Department of Mathematics, University of Manchester, Manchester.
Shampine, L. F. (1985). Interpolation for Runge-Kutta methods, *SIAM J. Numer. Anal.* **22**, 1014–1027.
Wheldon, T., Kirk, J. and Finlay, H. (1974). Cyclical granulopoiesis in chronic granulocytic leukemia: A simulation study., *Blood* **43**, 379–387.

APPENDIX 1 SPECIFICATION OF PROBLEMS

Problem Class A. *RDE with time dependent delays.*

A1: Model of blood production (Mackey and Glass, 1977)

$$\begin{aligned} y'(t) &= \frac{0.2y(t-14)}{1+y(t-14)^{10}} - 0.1y(t), \quad t_0 = 0, \quad t_f = 500, \\ \phi(t) &= 0.5, \quad \text{for} \quad t \leq 0. \end{aligned}$$

A2: Model of chronic granulocytic leukemia (Wheldon, Kirk and Finlay, 1974)

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

$$
\begin{aligned}
y_2'(t) &= \frac{100y_1(t)}{1+40y_2(t)} - 2.43y_2(t),\\
& \quad t_0 = 0, \quad t_f = 100,\\
\phi_1(t) &= 1.05767027/3, \quad \phi_2(t) = 1.030713491/3, \quad \text{for} \quad t \le 0.
\end{aligned}
$$

Problem Class B. *RDE with small time dependent delays.*

B1: (Neves, 1975)

$$
\begin{aligned}
y'(t) &= 1 - y\left(\exp\left(1 - 1/t\right)\right), \quad t_0 = 0.1, \quad t_f = 10,\\
\phi(t) &= \ln(t), \quad \text{for} \quad 0 < t \le 0.1.
\end{aligned}
$$

(analytical solution: $y(t) = \ln(t)$, vanishing delays: $t = 1$)

B2: (Neves and Thompson, 1992b)

$$
\begin{aligned}
y'(t) &= f(y(t/2)) - y(t), \quad t_0 = 0, \quad t_f = 2\ln(66),\\
y(0) &= 1,
\end{aligned}
$$

where $f(s) = 1$ if $s < 0$ and $f(s) = -1$ if $s \ge 0$. (analytical solution:

$$
y(t) = \begin{cases} 2\exp(-t) - 1, & \text{for} \quad 0 \le t \le 2\ln(2),\\ 1 - 6\exp(-t), & \text{for} \quad 2\ln(2) < t \le 2\ln(6),\\ 66\exp(-t) - 1, & \text{for} \quad 2\ln(6) < t \le 2\ln(66), \end{cases}
$$

vanishing delay: $t = 0$)

Problem Class C. *RDE with state dependent delays.*

C1: (Paul, 1994)

$$
\begin{aligned}
y'(t) &= -2y(t - 1 - |y(t)|)(1 - y^2(t)), \quad t_0 = 0, \quad t_f = 30,\\
\phi(t) &= 1/2.
\end{aligned}
$$

C2: (Paul, 1994)

$$
\begin{aligned}
y_1'(t) &= -2y_1(t - y_2(t)),\\
y_2'(t) &= \frac{|y_1(t - y_2(t))| - |y_1(t)|}{1 + |y_1(t - y_2(t))|},\\
& \quad t_0 = 0, \quad t_f = 40,\\
\phi_1(t) &= 1, \quad \phi_2(t) = 1/2, \quad \text{for} \quad t \le 0.
\end{aligned}
$$

C3: Model of hematopoiesis (Mahaffy, Bélair and Mackey, 1996)

$$
\begin{aligned}
y_1'(t) &= \hat{s}_0 y_2(t - T_1) - \gamma y_1(t) - Q,\\
y_2'(t) &= f(y_1(t)) - k y_2(t),
\end{aligned}
$$

$$
\begin{aligned}
y_3'(t) &= 1 - \frac{Qe^{\gamma y_3(t)}}{\hat{s}_0 y_2(t - T_1 - y_3(t))}, \\
& \quad t_0 = 0, \quad t_f = 300, \\
\phi_1(0) &= 3.325, \quad \phi_3(0) = 120, \\
\phi_2(t) &= \begin{cases} 10, & \text{for} \quad -T_1 \le t \le 0, \\ 9.5, & \text{for} \qquad t < -T_1, \end{cases}
\end{aligned}
$$

where $f(y) = \dfrac{a}{1 + Ky^r}$, $\hat{s}_0 = 0.0031$, $T_1 = 6$, $\gamma = 0.001$, $Q = 0.0275$, $k = 2.8$, $a = 6570$, $K = 0.0382$, $r = 6.96$.

C4: The same equation in C3 with $t_0 = 0$, $t_F = 100$,

$$
\begin{aligned}
\phi_1(0) &= 3.5, \quad \phi_3(0) = 50, \\
\phi_2(t) &= 10, \quad \text{for} \quad t \le 0,
\end{aligned}
$$

$\hat{s}_0 = 0.00372$, $T_1 = 3$, $\gamma = 0.1$, $Q = 0.00178$, $k = 6.65$, $a = 15600$, $K = 0.0382$, $r = 6.96$.

Problem Class D. ***RDE with small state dependent delays.***

D1: (Neves, 1975)

$$
\begin{aligned}
y_1'(t) &= y_2(t), \\
y_2'(t) &= -y_2(\exp(1 - y_2(t)))y_2^2(t)\exp(1 - y_2(t)), \\
& \quad t_0 = 0.1, \quad t_f = 5, \\
\phi_1(t) &= \ln(t), \quad \phi_2(t) = 1/t, \quad \text{for} \quad 0 < t \le 0.1.
\end{aligned}
$$

(analytical solution: $y_1(t) = \ln(t), y_2(t) = 1/t$, vanishing delay: $t = 1$)

D2: Model of antigen antibody dynamics with fading memory (Gatica and Waltman, 1982)

$$
\begin{aligned}
y_1'(t) &= -r_1 y_1(t) y_2(t) + r_2 y_3(t), \\
y_2'(t) &= -r_1 y_1(t) y_2(t) + \alpha r_1 y_1(t - y_4(t)) y_2(t - y_4(t)), \\
y_3'(t) &= r_1 y_1(t) y_2(t) - r_2 y_3(t), \\
y_4'(t) &= 1 + \frac{3\delta - y_1(t) y_2(t) - y_3(t)}{y_1(t - y_4(t)) y_2(t - y_4(t)) + y_3(t - y_4(t))} \exp(\delta y_4(t)), \\
& \quad t_0 = 0, \quad t_f = 40, \\
\phi_1(t) &= 5, \quad \phi_2(t) = 0.1, \quad \phi_3(t) = \phi_4(t) = 0, \quad \text{for} \quad t \le 0,
\end{aligned}
$$

where $r_1 = 0.02$, $r_2 = 0.005$, $\alpha = 3$, $\delta = 0.01$. (vanishing delay: $t = 0$)

Problem Class E. ***NDE with time dependent delays.***

E1: Model of food-limited population (Kuang and Feldstein, 1991)

$$\begin{aligned} y'(t) &= ry(t)(1 - y(t-1) - cy'(t-1)), \quad t_0 = 0, \quad t_f = 40, \\ \phi(t) &= 2 + t, \quad \text{for} \quad t \leq 0, \end{aligned}$$

where $r = \pi/\sqrt{3} + 1/20,\ c = \sqrt{3}/2\pi - 1/25$.
E2: Logistic Gauss-type predator-prey systems (Kuang, 1991)

$$\begin{aligned} y_1'(t) &= y_1(t)(1 - y_1(t-\tau) - \rho y_1'(t-\tau)) - \frac{y_2(t)y_1^2(t)}{y_1^2(t)+1}, \\ y_2'(t) &= y_2(t)\left(\frac{y_1^2(t)}{y_1^2(t)+1} - \alpha\right), \\ & \quad t_0 = 0, \quad t_f = 2, \\ \phi_1(t) &= 33/100 - t/10, \quad \phi_2(t) = 111/50 + t/10, \quad \text{for} \quad t \leq 0, \end{aligned}$$

where $\alpha = 1/10,\ \rho = 29/10,\ \tau = 21/50$.

Problem Class F. *NDE with small time dependent delays.*

F1: (Jackiewicz, 1981)

$$\begin{aligned} y'(t) &= 2\cos(2t)y^{2\cos(t)}(t/2) + \ln(y'(t/2)) - \ln(2\cos(t)) - \sin(t), \quad t_0 = 0, \quad t_f = 1, \\ \phi(0) &= 1, \quad \phi'(0) = 2. \end{aligned}$$

(analytical solution: $y(t) = \exp(\sin(2t))$, vanishing delay: $t = 0$)
F2: (Neves and Thompson, 1992b)

$$\begin{aligned} y'(t) &= y'(2t - 1/2), \quad t_0 = 0.25, \quad t_f = 0.499, \\ \phi(t) &= \exp(-t^2), \quad \phi'(t) = -2t\exp(-t^2), \quad \text{for} \quad t \leq 0.25. \end{aligned}$$

(analytical solution: $y(t) = y_i(t) = \exp(-4^i t^2 + B_i t + C_i)/2^i + K_i$, for $t \in [x_i, x_{i+1}]$, where

$$\begin{aligned} x_i &= (1 - 2^{-i})/2, \\ B_i &= 2(4^{i-1} + B_{i-1}), \\ C_i &= -4^{i-2} - B_{i-1}/2 + C_{i-1}, \\ K_i &= -\exp(-4^i x_i^2 + B_i x_i + C_i)/2^i + y_{i-1}(x_i), \end{aligned}$$

with $B_0 = C_0 = K_0 = 0$, vanishing delay: $t = 1/2$)
F3:

$$\begin{aligned} y'(t) &= \exp(-y(t)) + L_3\left[\sin(y'(\alpha(t))) - \sin\left(\frac{1}{3+\alpha(t)}\right)\right], \quad t_0 = 0, \quad t_f = 10, \\ \phi(0) &= \ln(3), \quad \phi'(0) = 1/3, \end{aligned}$$

where $\alpha(t) = 0.5t(1 - \cos(2\pi t))$, $L_3 = 0.2$. (analytical solution: $y(t) = \ln(t+3)$, vanishing delays: $t = 0, (2i-1)/2$, for $i = 1, 2, \cdots, 10$)
F4: The same equation in F3 with $L_3 = 0.4$.
F5: The same equation in F3 with $L_3 = 0.6$.

Problem Class G. *NDE with state dependent delays.*
(The problems are primarily of theoretical interest only. From a numerical point of view a method should be able to solve them reliably on the prescribed range.)

G1: (El'sgol'ts and Norkin, 1973, p. 44)

$$\begin{aligned} y'(t) &= -y'(t - y^2(t)/4), \quad t_0 = 0, \quad t_f = 1, \\ \phi(t) &= 1 - t, \quad \text{for} \quad t \leq 0, \quad \phi'(t) = -1, \quad \text{for} \quad t < 0. \end{aligned}$$

(analytical solution: $y(t) = t + 1$)
G2: (El'sgol'ts and Norkin, 1973, pp. 44–45)

$$\begin{aligned} y'(t) &= -y'(y(t) - 2), \quad t_0 = 0, \quad t_f = 1, \\ \phi(t) &= 1 - t \quad \text{for} \quad t \leq 0, \quad \phi'(t) = -1, \quad \text{for} \quad t < 0. \end{aligned}$$

(analytical solution: $y(t) = t + 1$)

Problem Class H. *NDE with small state dependent delays.*

H1: (Castleton and Grimm, 1973)

$$\begin{aligned} y'(t) &= \frac{-4ty^2(t)}{4 + \ln^2(\cos(2t))} + \tan(2t) + \frac{1}{2}\arctan\left(y'\left(\frac{ty^2(t)}{1 + y^2(t)}\right)\right), \quad t_0 = 0, \quad t_f = 0.9\pi/4, \\ \phi(0) &= 0, \quad \phi'(0) = 0. \end{aligned}$$

(analytical solution: $y(t) = -\ln(\cos(2t))/2$, vanishing delay: $t = 0$)
H2: (Hayashi, 1996)

$$\begin{aligned} y'(t) &= \cos(t)(1 + y(ty^2(t))) + L_3 y(t) y'(ty^2(t)) \\ &\quad + (1 - L_3)\sin(t)\cos(t\sin^2(t)) - \sin(t + t\sin^2(t)), \\ &\quad t_0 = 0, \quad t_f = \pi, \\ \phi(0) &= 0, \quad \phi'(t) = 1, \end{aligned}$$

where $L_3 = 0.1$.
(analytical solution: $y(t) = \sin(t)$, vanishing delays: $t = 0, \pi/2, \pi$)
H3: The same equation in H2 with $L_3 = 0.3$.
H4: The same equation in H2 with $L_3 = 0.5$.

DISCUSSION

Speaker : W. Enright

R. Hanson : How is this work to be distributed or made available to the user community?

W. Enright : We are not yet distributing the DDETEST package as we are still adding new problems and deleting them. We will soon be distributing it freely to those interested (via FTP).

B. Smith : Of all the test problems, can you identify a "difficult subset" or a parametrised set which can be used as a benchmark to quantify the potential of new methods that may be developed in the future?

W. Enright : This would be an important contribution. We don't have enough experience yet to identify such a subset of our problems. Many of our problems are members of a parametrised set and others are chosen to stretch the limits of some aspects of existing methods.