Software for Ordinary and Delay Differential Equations: Accurate Discrete Approximate Solutions are not Enough *

W.H. Enright
Department of Computer Science
University of Toronto
Toronto, Canada M5S 3G4

enright@cs.utoronto.ca

Abstract

Numerical methods for both ordinary differential equations (ODEs) and delay differential equations (DDEs) are traditionally developed and assessed on the basis of how well the accuracy of the approximate solution is related to the specified error tolerance on an adaptively-chosen, discrete mesh. This may not be appropriate in numerical investigations that require visualization of an approximate solution on a continuous interval of interest (rather than at a small set of discrete points) or in investigations that require the determination of the 'average' values or the 'extreme' values of some solution components.

In this paper we will identify modest changes in the standard error-control and stepsize-selection strategies that make it easier to develop, assess and use methods which effectively deliver approximations to differential equations (both ODEs and DDEs) that are more appropriate for these type of investigations. The required changes will typically increase the cost per step by up to 40%, but the improvements and advantages gained will be significant. Numerical results will be presented for these modified methods applied to two example investigations (one ODE and one DDE).

Keywords: Runge-Kutta methods, Delay differential equations, Ordinary differential equations, Interpolation.

AMS Subject Classifications: 65L05, 65L10

Abbreviated Title: Software for DDEs

*This research was supported by the Natural Science and Engineering Research Council of Canada.

1 Introduction

If numerical methods for ordinary differential equations (ODEs) are to be widely used, they should meet the broadest expectations of potential users. Users, in many cases, are interested in visualization and/or investigations of properties of the solution, not simply the approximation at discrete mesh points. They are likely to be working in a problem solving environment (PSE) with a choice of methods available. The implications for developers of ODE software are significant and include the following implementation issues:

- The method should adopt a standard, easy to understand, interpretation of the error control mechanism(s) that are available and a standard program interface (calling sequence).
- The options and additional parameters (if required) should be specified in a consistent fashion for all methods. This includes stepsize constraints and accuracy specification.
- The method should adopt a standard representation of the approximate solution. (For example, a vector of piecewise polynomials.)

In recent years there has been considerable progress made addressing these implementation issues as new methods have been introduced and extra features provided to make the current generation of ODE software more accessible to a wider audience of users. For example we have implemented a family of ODE methods and associated software tools ([6], [3], [1] and [5]) and Shampine and his colleagues ([13], [9], [14] and [12]) have implemented, in the MATLAB PSE [10], a similar family of ODE methods. One way to view the progress that has been made is to interpret it as a natural evolution of 'adaptive' ODE methods, starting with a classical fixed stepsize method and introducing three features,

- Variable stepsize discrete approximations,
  (with maximum stepsize, h):

  $$\{x_i, y_i\}_{i=0}^N, \ \ \max_{i=1}^{N} |y(x_i) - y_i| = O(h^p).$$

  [Such methods are typically based on an underlying discrete Runge-Kutta (RK) formula (or a formula pair). Effective classes of formula pairs of this type include, for example, those derived by Fehlberg [7], Prince and Dormand [11], and Verner [15].]

- Variable stepsize continuous extensions, $S(x)$,
  (for visualization):

  $$S(x_i) = y_i, \ \ ||S(x) - y(x)|| = O(h^p), \ \text{ for } \ x \in [x_0, x_N].$$

  [Note that if the underlying discrete formula of such a method is a RK formula, such a continuous approximation is referred to as a continuous RK formula (CRK).]

- Variable stepsize CRKs with direct defect error control,
  (to obtain 'tolerance proportionality' and a generic convergence result):
  That is, the interpolant $S(x)$ satisfies,

  $$||S'(x) - f(x, S(x))|| \leq TOL,$$

  which implies,

  $$\begin{aligned} ||S(x) - y(x)|| &\leq K_1 TOL \\ ||S'(x) - y'(x)|| &\leq K_2 TOL. \end{aligned}$$

  [Note that $\delta(x) \equiv S'(x) - f(x, S(x))$ is defined to be the defect of the approximate solution and 'direct defect control' refers to the use of error control strategies that attempt to correctly estimate and bound the dominant term in the asymptotic expansion of this defect.]

In this paper we are concerned with making the new generation of ODE methods, in particular RK methods for DDEs, more accessible and usable by scientists and engineers in a wide variety of research areas. We are particularly interested in showing that this can be done without a significant increase in cost. Although we will use numerical methods that we have implemented to make this point, we do not want to infer that other approaches are not equally valid or that or our methods are optimal – only that our objective can be achieved. We use two examples to illustrate and motivate our focus. The first involves a standard initial value problem (IVP) and is chosen because the lessons learned and implications discussed also apply directly to DDEs.

1.1 An example IVP investigation

Consider a typical use of an IV method in a PSE, where a predator-prey relationship is modeled by the IVP:

$$y_1' = y_1 - 0.1 y_1 y_2 + 0.02x$$

$$y_2' = -y_2 + 0.02 y_1 y_2 + 0.008x$$

with specified initial conditions. For example, for most of the tests reported here we use,

$$y_1(0) = 30, \;\; y_2(0) = 20.$$

In this application $y_1(x)$ represents the 'prey' population at time x and $y_2(x)$ represents the 'predator' population at time x. We know that solutions to this problem exhibit oscillatory behaviour as x increases. A biologist may be interested in whether the solution components of this equation are 'almost periodic' (in the sense that the difference between points where successive maximums occur is constant) and whether the local maximum values approach a steady state exponentially (see figure 1).

Figure 1 illustrates qualitatively the objectives of such an investigation for a typical set of initial conditions (in this case $y_1(0) = 160$) where the interval of interest is $[0, 140]$ and the prey population is plotted vs x. The displayed solution is an accurate approximate solution generated using a reliable numerical method with a stringent accuracy request (10^{-10} in this case). This is also the technique used to generate the 'true' solution for all our numerical tests. We tried different reliable methods at different accuracy requests and always obtained consistent results.

In figure 1 (as in most of the figures in this paper) the horizontal axis represents the independent variable, x, and the vertical axis represents various components of the dependent variable(s), $y(x)$.

To answer the questions posed by this investigation the numerical method should be capable of providing an appropriate 'visualization' of the prey population on the interval of interest. Three techniques that can be used to provide such a visualization are illustrated in figures 2 - 4 where the underlying IV method is ode45 of MATLAB [10] and the error control option used is $ATOL = TOL = 10^{-10}$. Note that the first two techniques do not deliver effective visualizations and would not be particularly suitable for locating the local extrema. (Although figure 3 and figure 4 are similar, a closer detailed inspection of the associated errors, not reported here, revealed the off-mesh error associated with figure 3 to be much greater than that associated with figure 4. In particular the location of the fourth and fifth local maximum of the prey population is not adequate to answer the question that is being investigated.) On the other hand the third technique (see figure 4) provides sufficient accuracy for the question to be answered over a range of accuracy requests. Various methods using this technique will deliver consistent visualizations.

To see this we show, in figure 5, an alternative visualization of an approximate solution to this problem determined by an 8^{th} order CRK method (using an equivalent accuracy request). This visualization is virtually indistinguishable from that corresponding to ode45 (figure 4). The robustness and quality of the visualizations associated with this technique makes it particularly appropriate for use in a PSE.

1.2 An essential requirement

For these ODE methods we can insist that an additional essential requirement be satisfied. We will first investigate this requirement in the context of IVPs and then consider the implication for DDEs. We consider the standard IVP case first as the implications for methods designed for this less complicated class of problems also apply to the more general class of DDE problems. This essential requirement is,

- When applied to:

$$y' = f(x, y),\ y(a) = y_0,\ \text{on}\ \ [a, b],$$

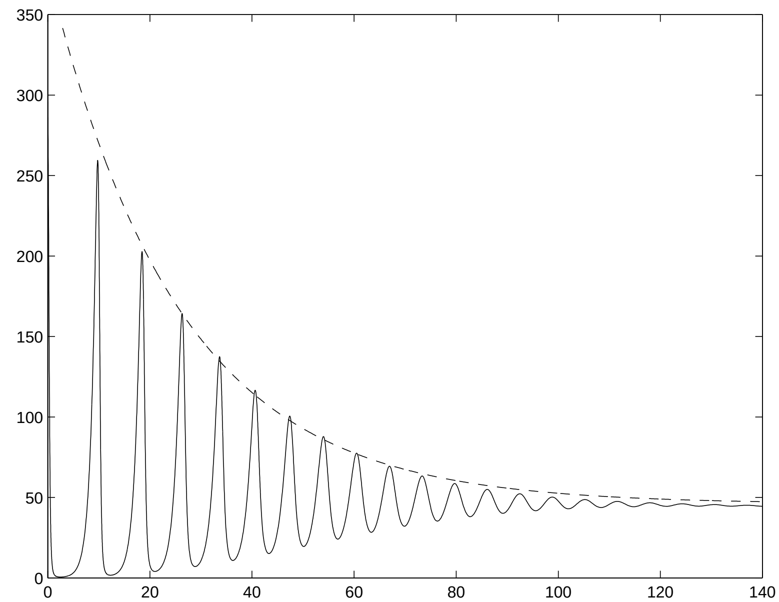
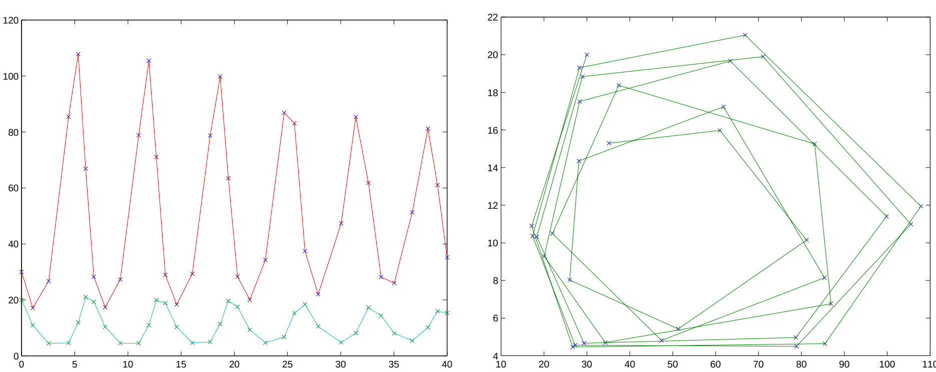


Figure 1: Do the local maximum of the prey population decay exponentially? Is the time between the local maximum almost constant?



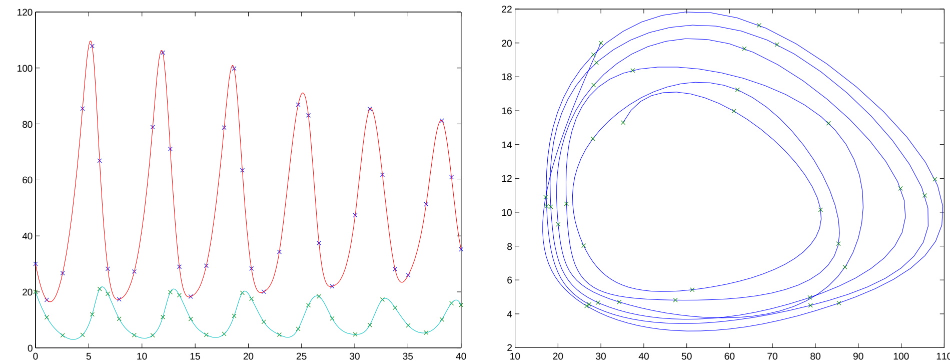
(a) Solution Plot

(b) Phase Portrait ($y_2(x)$ vs $y_1(x)$)

Figure 2: Visualizing the solution of the predator-prey problem with ode45 using only the discrete approximation and its piecewise linear interpolant

with a specified accuracy, TOL, the method generates a piecewise polynomial, $S(x)$, defined for $x \in [a,b]$ satisfying,

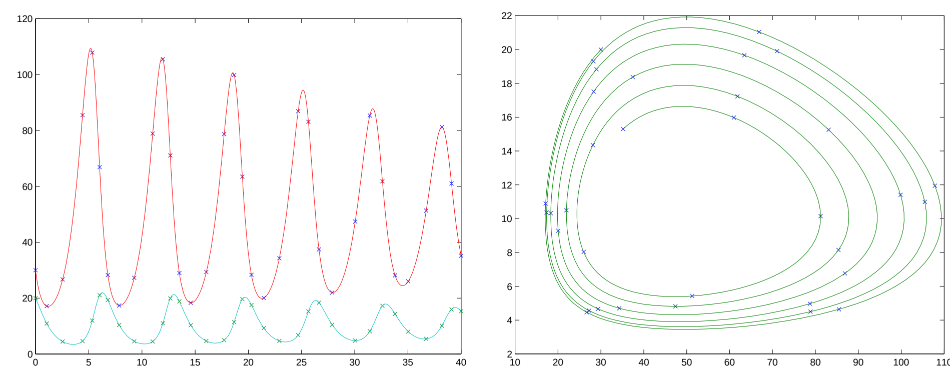
$$\|S(x) - y(x)\| \leq K_M TOL.$$



(a) Solution Plot

(b) Phase Portrait ($y_2(x)$ vs $y_1(x)$)

Figure 3: Visualizing the solution of the predator-prey problem with ode45 using a standard cubic spline approximation to the discrete solution



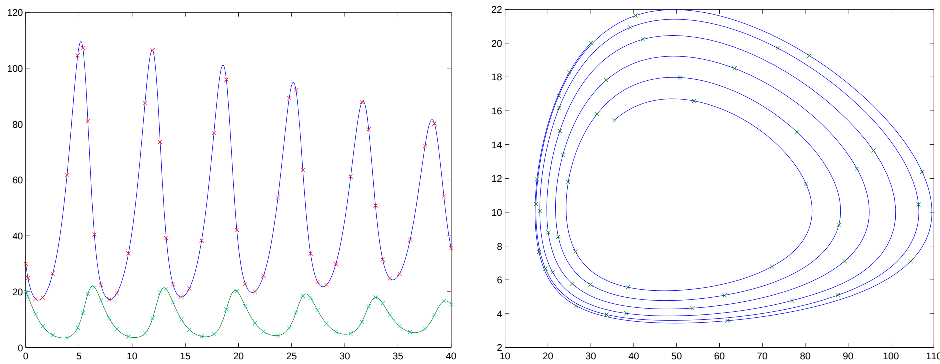
(a) Solution Plot

(b) Phase Portrait ($y_2(x)$ vs $y_1(x)$)

Figure 4: Visualizing the solution of the predator-prey problem with ode45 using its associated CRK, S(x)

Note that K_M can depend on the method and the problem and can be interpreted as the 'numerical condition number' associated with method M applied to the problem. With an appropriate choice of error and stepsize control (For example, direct defect control) we can ensure that K_M will be almost independent of the method.

Let $z_i(x)$ be the solution of the local IVP on step i and consider the approximation, $\{x_i, y_i\}_{i=0}^{N}$, associated with a p^{th}- order discrete RK formula. We will



(a) Solution Plot

(b) Phase Portrait ($y_2(x)$ vs $y_1(x)$)

Figure 5: Visualizing the solution of the predator-prey problem using an 8^{th} order CRKand its associated CRK, S(x)

consider three types of CRKs that can be associated with this discrete RK formula. The associated interpolants, $S(x)$, $\bar{S}(x)$, and $\tilde{S}(x)$, (of increasing cost and reliability) are characterized below. Each will satisfy this essential requirement for $x \in [x_{i-1}, x_i]$:

Type I: $S(x) = z_i(x) + O(h^p)$ with the leading term in the asymptotic expansion of the defect, satisfying

$$\delta(x) = d(f)h^{p-1} + O(h^p),$$

with $d(f)$ depending on the problem and the method.

Type II: $\bar{S}(x) = z_i(x) + O(h^{p+1})$ with,

$$\bar{\delta}(x) = \bar{d}(f)h^p + O(h^{p+1}),$$

with $\bar{d}(f)$ depending on the problem and the method.

Type III: $\tilde{S}(x) = z_i(x) + O(h^{p+1})$ with,

$$\tilde{\delta}(x) = \tilde{d}(f)h^p + O(h^{p+1}),$$

with $\tilde{d}(f)$ depending only on the problem.

1.3 Error and stepsize Control

With CRK methods satisfying this essential requirement, one can monitor the magnitude of the defect associated with each step and accept the step only if an estimate of this quantity is less than the error tolerance, TOL.

It should be observed that methods that use such defect control strategies cannot employ local extrapolation, either to estimate the magnitude of the defect or improve the accuracy of the accepted solution. (Any attempt to do so would require a new, more expensive CRK and/or a more expensive estimate of the associated defect.) Note also that, with direct defect control, one can derive associated estimates $\delta(x)$, $\bar{\delta}(x)$ that have been found (through extensive numerical tests) to be reliable on each step 'with high probability' (typically based on sampling the defect at one or two points per step). In addition, estimates of $\tilde{\delta}(x)$ that are reliable (asymptotically justified) for all steps, can readily be developed.

With direct defect control one can prove (see for example [2]) the desired convergence result:

$$\|S(x) - y(x)\| \leq K_M TOL,$$

where K_M depends primarily on the problem.

We have implemented IVP, BVP and DDE methods (see [3], [1] for details) based on this approach. These methods provide, as an option to the user, the choice of either a continuous extension of type $\bar{S}(x)$ or the more reliable but more expensive $\tilde{S}(x)$. The DDE method of this family is ddverk [1] and it is available through NETLIB (www.netlib.org/index.html).

Table 1 identifies the cost per step in terms of derivative evaluations required for the different types of CRKs associated with some effective p^{th}-order discrete RK formulas. We include in our reporting of cost the additional derivative evaluation required when using defect control and either $\bar{S}(x)$ or $\tilde{S}(x)$ (with indirect local error control, $S(x)$, no derivative evaluations are required to determine the local error estimate). We do not claim that these formulas are optimal, but rather that they illustrate that the additional costs of the more reliable interpolating schemes are not prohibitive. The first two CRK formulas are based on discrete RK formulas derived in [7], with the corresponding CRK proposed in [6]. The next three CRK formulas are obtained from software developed for the derivation of discrete formula pairs in [15] with the corresponding CRK derived using the algorithm given in [16]. In this table, s is the number of stages necessary to determine an associated $S(x)$; $(\bar{s} - 1)$ is the number of stages necessary to determine an associated $\bar{S}(x)$; and $(\tilde{s} - 1)$ is the number of stages necessary to determine an associated $\tilde{S}(x)$. This table quantifies the 'extra cost' associated with the more reliable defect control strategies.

For any DDE method, an associated off-mesh interpolation technique (to evaluate the delayed solution approximations on step i, for $x < x_{i-1}$,) must be part of the overall method, although this technique does not have to correspond to $S(x)$. Note also that this table applies to both IVP and DDE methods based on the respective p^{th}-order, discrete formula and it identifies the 'cost' of forming the underlying interpolant and estimating the associated defect.

Discrete Formula	p	s	$\bar{s}$	$\tilde{s}$
CRK4	4	4	6	7
CRK5	5	7	9	11
CVSS6B	6	9	11	14
CVSS7	7	11	15	20
CVSS8	8	15	21	28
ode45	5	7	9	11

Table 1: Cost per step of some typical CRK ODE methods

1.4 Three versions of ode45:

We have modified ode45 so the user has some choice in the selection of the type of CRK and associated error control. By setting the parameter eropt, one of three error control strategies (and associated interpolant) is used:

eropt = I: Indirect local error control– using $S(x)$. This gives the identical results that the built-in routine provides at a cost of (see table 1) seven derivative evaluations per step.

eropt = II: Direct defect control – using $\bar{S}(x)$. The error estimate is based on a single sampled evaluation of the defect and has a high probability of being reliable. The associated cost is nine derivative evaluations per step.

eropt = III: Strict direct defect control – using $\tilde{S}(x)$. The leading term in the expansion of this defect is reliably estimated and the cost is eleven derivative evaluations per step.

There is clearly a cost/reliability trade off to be considered when selecting the error control option for a particular application.

We now present numerical results for the predator-prey investigation for these three versions of ode45. In each case we invoked the method and, after each step, checked the sign of the derivative of the interpolant (at x_{i-1} and x_i) to see if the integration had passed through a local maximum of the prey population. If it had, we then determined the location and value of this local maximum (by solving for the zeros of the derivative of the interpolant – the zeros of $S'(x)$). After the completion of the integration a MATLAB linear least squares method was used to determine how well the observed computed data, (the locations and values of the local maximums), fit the hypothesis that the solution was almost periodic and that the magnitude of the local maximum of the prey population decayed to its steady state value at an exponential rate.

That is, if the set of local maximums identified by this technique is $\{\hat{x}_j, \hat{y}_j\}_{j.=1}^{M}$, then a linear least squares solver was used to determine the 'best' exponential fit of the form,

$$\ln(\hat{y}_j) \approx a\,\hat{x}_j \;+\; b.$$

The corresponding value, b, was then compared to the 'true' value (associated with the data), which was precomputed using an accurate approximation to

$y(x)$. This accurate approximation to $y(x)$ was also used to determine the reported measures of error: ger, ymerr and experr. Similarly a best least squares fit of the form, $\hat{x}_j - \hat{x}_{j-1} \approx R$, for $j = 2, 3 \cdots M$ was determined.

Note that the technique we have used to allow the biologist to explore the validity of his/her hypothesis is only one generic approach that could be used. Other equally effective techniques could be used, particularly if the underlying numerical method allows a user to specify 'events' or 'g-stops' (see, for example, [12]).

Table 2 reports the following statistics:

steps: The number of time steps.

fcn: The number of derivative evaluations.

ger: The maximum magnitude of the error in the solution, measured in units of TOL.

ymerr: The maximum magnitude of the error in the identified local maximums (of the prey population), measured in units of TOL.

experr: The error in the reported value of the 'best' exponential fit to the decay exhibited by the mathematical model, measured in units of TOL.

R(res): Best least square fit and residual for determining whether the prey population is almost periodic.

These results show that the type II and type III CRK versions of ode45 provide consistent and accurate answers to the questions the biologist is investigating (over a range of accuracy requests) while the type I version does not do so at the most relaxed error tolerance ($TOL = 10^{-2}$).

2 General purpose DDE software

As for IVP software, a new generation of numerical methods for DDEs is now possible and is being developed. These new methods address the issues we have identified for IVPs as well as other issues that are particular to DDEs.

Any method that can be applied to DDE problems with multiple delays, and both retarded and neutral delays must inherently have a complex calling sequence just to specify the 'mathematical' problem,

$$\begin{aligned} y' &= f(x, y(x), y(x-\sigma_1) \cdots y(x-\sigma_k), \\ &\quad y'(x-\sigma_{k+1}), \cdots y'(x-\sigma_{k+\ell})) \end{aligned}$$

where

$$y(x) = \phi(x), \quad y'(x) = \phi'(x), \quad \text{for} \quad x \leq x_0,$$

and

$$\sigma_i \equiv \sigma_i(x, y(x)) \geq 0 \quad \text{for} \quad i = 1, 2 \cdots k + \ell.$$

eropt	TOL	10^{-2}	10^{-4}	10^{-6}
I	steps	71	148	367
	fcn	511	961	2239
	ger	30.	8.3	3.9
	ymerr	12.	1.1	2.2
	experr	24.8	2.9	5.6
	R(res)	6.43 (.4)	6.37 (.05)	6.37 (.05)
II	steps	92	184	397
	fcn	921	1769	3385
	ger	4.1	2.3	4.3
	ymerr	.70	1.1	3.5
	experr	2.2	1.7	5.4
	R(res)	6.36 (.07)	6.37 (.05)	6.37 (.05)
III	steps	92	185	408
	fcn	1171	2131	4441
	ger	1.5	1.7	2.6
	ymerr	.78	.82	2.2
	experr	2.7	1.7	3.7
	R(res)	6.36 (.06)	6.37 (.05)	6.37 (.05)

Table 2: Results for the 3 versions of ode45 on the predator-prey investigation

To specify this problem the user must supply, in addition to the subroutine to evaluate the differential equation and the range of integration, $[a, b]$,

- subroutines to evaluate each $\sigma_i(x, y)$.
- subroutines to evaluate $\phi(x)$, and $\phi'(x)$.

With this generality in specifying the mathematical problem, difficulties can arise with well defined mathematical problems that are inherently expensive to approximate numerically. Examples of such problems are those that involve multiple state-dependent delays and neutral problems that do not satisfy,

$$x \leq \bar{x} \;\; \Rightarrow \;\; x - \sigma(x, y(x)) \leq \bar{x} - \sigma(\bar{x}, y(\bar{x})).$$

2.1 An example DDE investigation

Consider a typical application, similar to our IVP example, based on a model of the spread of an infectious disease (this is a well-known problem discussed, for example, in [8] p.295).

Let y_1 represent the susceptible portion of the population, y_2 represent the infected portion of the population and y_3 represent the immunized portion of the population. Assume that the immunized group becomes susceptible after

10 units of time and that there is an incubation period of 1 unit. The resulting DDE is,

$$
\begin{aligned}
y_1' &= -y_1(x)y_2(x-1) + y_2(x-10), \\
y_2' &= y_1(x)y_2(x-1) - y_2(x), \\
y_3' &= y_2(x) - y_2(x-10),
\end{aligned}
$$

with a typical set of initial functions,

$$y_1(x) = 5, \;\; y_2(x) = 0, \;\; y_3(x) = 1 \;\; \text{for} \;\; x \leq 0.$$

The solution to this problem has a discontinuity in $y'(x)$ at the initial point, $x = 0$, and subsequent discontinuities in higher order derivatives at $x = 1$, $x = 2 \cdots$; $x = 10$, $x = 11$, $x = 20$ $\cdots$.

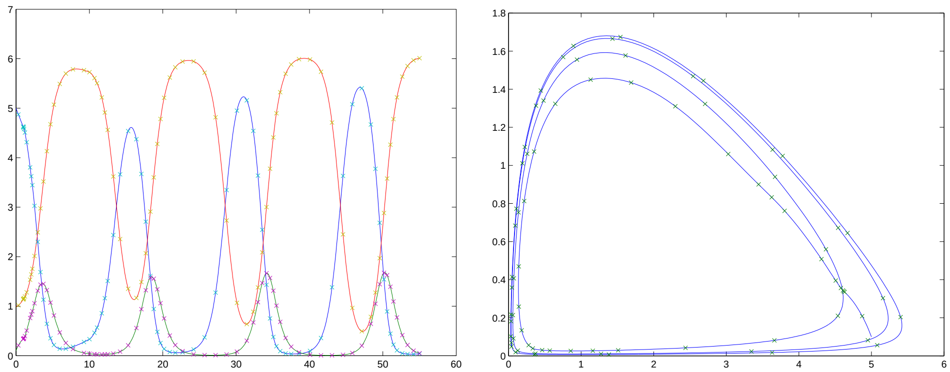
One may be interested in investigating the local extrema of some of the populations. We will consider the use of ddverk which provides two different choices for a Type II CRK (IIa corresponds to using one sampled point per step to estimate the magnitude of the defect while IIb corresponds to using two sampled points) and one Type III CRK.

Similar to our IVP investigation, our numerical investigation was accomplished by checking the sign of the derivative approximations at the discrete points, x_i and signaling when a local maximum had been passed (that is, when the value of the approximation $S'(x)$ had changed sign). At that point a bisection search was invoked to accurately determine the location and value of the associated local maximum. It is worth noting that, if one only wanted the location and corresponding value of the local extrema, then the extra cost of evaluating $S(x)$ on a fine mesh (for visualization) for all components on the whole interval would not be necessary. This could lead to additional savings in computer time and storage.

Table 3 reports the performance of the three versions of ddverk on this infectious disease investigation. Each version provides a consistent and hopefully accurate answer to the location and value of the local extrema of the infected population (see figures 6 and 7). Note that we should expect to observe tolerance proportionality only over a reasonable range of accuracy requests. For 'relaxed' values of TOL the asymptotic analysis that justifies our strategies will not be applicable while, for stringent values of TOL the contribution of roundoff error may dominate the truncation error (and not be proportional to TOL). [Note that the results reported here are for the version of ddverk available through NETLIB. This version is known not to have full machine accuracy in the coefficients defining the CRK formula. To invesigate the effect this might have, we have run a modified version of ddverk, where these coefficients are accurate to full machine precision, and observed very little change.]

3 Summary and conclusions

We have shown that numerical methods for DDEs that are more appropriate for use in a PSE can be developed at a modest increase in cost. These methods



(a) Solution Plot

(b) Phase Portrait ($y_2(x)$ vs $y_1(x)$)

Figure 6: Visualizing the solution of the infectious disease problem using $\bar{S}(x)$ with $TOL = 10^{-4}$ and eropt = IIa

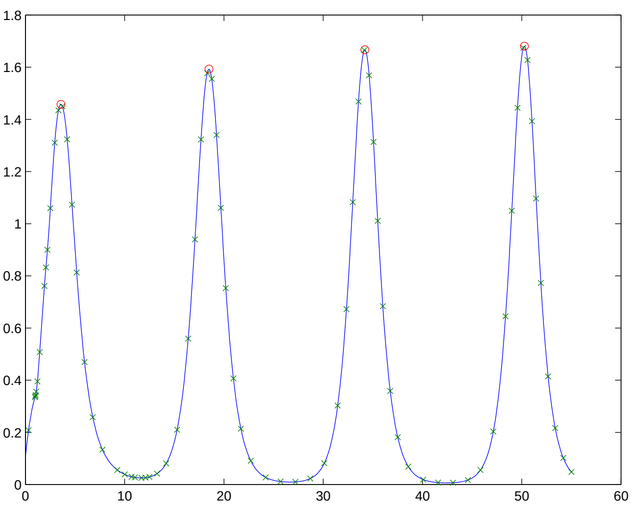


Figure 7: The local maximums of the infected population determined using $\bar{S}(x)$ with $TOL = 10^{-4}$ and eropt = IIa

will, when based on an underlying p^{th}-order discrete RK formula, produce an approximate solution as a piecewise polynomial with an accuracy that is uniformly p^{th} order for all x in the interval of interest. With direct defect control, these methods will deliver approximate solutions exhibiting tolerance proportionality with the 'constant of proportionality' depending primarily on the mathematical conditioning of the problem. Another advantage of these methods is that one

eropt	TOL	10^{-2}	10^{-4}	10^{-6}	10^{-8}
IIa	steps	44	76	140	280
	fcn	886	1418	2090	3653
	ger	5.5	108	158	168
	ymer	.14	.84	.84	180
IIb	steps	45	79	143	290
	fcn	992	1700	2357	4177
	ger	.78	71	102	96
	ymer	.34	.51	1.2	94
III	steps	46	80	153	301
	fcn	1101	1897	2898	5035
	ger	.75	41	53	69
	ymer	.15	.39	1.1	69

Table 3: Performance of different versions of ddverk on the infectious disease investigation

obtains accurate approximations to both the solution and its derivative.

The next step required to make these methods easier to use in a PSE involves the adoption of a simple hierarchical interface and the generation of several worked out examples (case studies) of typical investigations so potential new users will be able to quickly and painlessly try out the method. It is not enough to report the results of a few typical applications (as we have done here), but listings of actual drivers used to carry out these investigations are needed. These can be used as 'templates' for new users and often are the best way for a user to understand how to use existing software to solve a new problem. We are currently producing such examples.

References

[1] W.H. Enright and H. Hayashi, A delay differential equation solver based on a continuous Runge-Kutta method with defect control, Numerical Algorithms, 16, 1997, pp. 349-364.

[2] W.H. Enright and H. Hayashi, Convergence analysis of the solution of retarded and neutral delay differential equations by continuous methods, SIAM J. Numer. Anal., 35, 2, 1998, pp.572-585.

[3] W.H. Enright and P.H. Muir, A Runge-Kutta type boundary value ODE solver with defect control, SIAM J. Sci. Comp., 17, 1996, pp.479-497.

[4] W.H. Enright, The relative efficiency of alternative defect control schemes for high order Runge-Kutta formulas, SIAM J. Numer. Anal., 30, 5, 1993, pp. 1419-1445.

[5] W.H. Enright, Continuous numerical methods for ODEs with defect control, JACM, 125, 2001, pp. 159-170.

[6] W.H. Enright, K.R. Jackson, S.P. Nørsett and P.G. Thomsen, Interpolants for Runge-Kutta formulas, ACM Trans. Math. Soft., 12, 1986, pp.193-218.

[7] E. Fehlberg, Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle, Computing, 4, 1968, pp. 93-106.

[8] E. Hairer, S.P. Norsett and G. Wanner, Solving Ordinary Differential Equations I – Nonstiff Problems, Springer Verlag, Berlin, 1987.

[9] J. Kierzenka and L.F. Shampine, A BVP solver based on residual control and the MATLAB PSE, ACM Trans. Math. Soft., 27, 2001, pp. 299-316.

[10] The MathWorks, Inc., MATLAB 6.0, Natick MA, 2000.

[11] P. J. Prince and J. R. Dormand, High order Embedded Runge-Kutta Formulae, J. Comput. Appl. Math., 7, 1981, pp. 67-75.

[12] L.F. Shampine, I. Gladwell and S. Thompson, *Solving ODEs in in the MATLAB Problem Solving Environment*, Cambridge University Press, Cambridge, 2003.

[13] L.F. Shampine and M.W. Reichelt. The MATLAB ODE suite, SIAM J. Sci. Comp., 18, 1997, pp.1-22.

[14] L.F. Shampine and S. Thompson, Solving DDEs in MATLAB, Appl. Numer. Math., 37, 2001, pp. 441-458.

[15] J. H. Verner, Explicit Runge-Kutta Methods with Estimates of the Local Truncation Error, SIAM J. Numer. Anal.,15, 1978, pp 772-790.

[16] J. H. Verner, Differentiable Interpolants for High-order Runge-Kutta Methods, SIAM J. Numer. Anal., 30,5, 1993, pp. 1446-1466.