Numerical Solution of the Delay Differential Equations Using Explicit Runge-Kutta Methods

Basim N. Abood
University of Technology, Applied Science Department

Abstract

We investigate the stability of a numerical solution of the delay differential equation by using explicit Runge-Kutta method, where the delay has been approximated by using Lagrange and Hirmate interpolation polynomials.

Introduction

The model problem, which we considered, has the form:

$$y'(t) = f(t, y(t), y(t - d(t, y(t)))) \qquad (1)$$

where $d(t, y(t)) \geq 0$ is referred to as the "delay" and $t - d(t, y(t))$ is referred to as the "lag". In general, the delay is a function of both of t and the solution $y(t)$.

In our research, we use various types of Runge-Kutta methods to find a numerical solution for eq.(1).

Previous work on numerical methods for delay differential equations done by Paul [1, 2] and Al Mutib [3].

Basic Explicit Runge-Kutta Method:

An s-stages Runge-Kutta formula for eq.(1) to compute the numerical solution at $t_n + h_n$ is defined by.

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i K_i$$

$$K_i = f(t_n + c_i h_n, y_i) \quad i = 1, 2, \ldots, s$$

$$y_i = y_n + h_n \sum_{j=1}^{i-1} a_{ij} K_j$$

where a_{ij}, b_i and c_i are the coefficients of the Runge-Kutta formula and y_i is the associated approximation with $y(t_n)$.

The Runge-Kutta formula is usually represented by a table:

c_1	a_{11}	a_{12}	…	a_{1s}
c_2	a_{21}	a_{22}	…	a_{2s}
⋮	⋮	⋮	⋱	⋮
c_s	a_{s1}	a_{s2}	…	a_{ss}
	b_1	b_2	…	b_s

Where:

$$c_i = \sum_{j=1}^{s} a_{ij}, \quad \sum_{i=1}^{s} b_i = 1$$

Definition

A Runge-Kutta process is said to be explicit if $a_{ij} = 0$, $j \geq i$, is said to be semi- explicit if $a_{ij} = 0$, $j > i$ and is said to be implicit otherwise. In this paper, we are interested only in the second and fourth order Runge-Kutta formulas, which have the following forms and tables:

0	0	0
1	1	0
	½	½

$$K_1 = f(t_n, y_n)$$

$$K_2 = f(t_n + h, y_n + K_1)$$

$$y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2), \quad n = 1, 2, \ldots$$

0	0	0	0	0
½	½	0	0	0
½	0	½	0	0
1	0	0	1	0
	1/6	2/6	2/6	1/6

$$K_1 = f(t_n, y_n)$$

$$K_2 = f(t_n + \frac{h}{2}, y_n + \frac{K_1}{2})$$

$$K_3 = f(t_n + \frac{h}{2}, y_n + \frac{K_2}{2})$$

$$y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \quad n = 1, 2, \ldots$$

Solution of DDE using explicit Runge-Kutta method:

To find the numerical solution for eq.(1), we can classify it into two types:

1. The problem with ignore delay term (i.e., $d(t, y(t)) = 0$) which is a first order ordinary differential equations, and the analytic solution can be made directly.
2. The problem with constant delay.
3. The problem with variable delay.

In cases (2) and (3), we need to evaluate $y(t - d(t, y(t)))$ and since the Runge-Kutta methods produce approximate results only at the mesh points, and the approximate solution between the mesh point is a matter of interpolation. Besides the other good qualities of this method, Lagrange or Hermite interpolation between mesh points provides a numerical solution just accurate as the solution of mesh points.

When using an interpolation scheme to evaluate the delay term, one must a ware about the following:

1. The number of solution values to be retained at any one time.
2. Selection of the points for interpolation and which interpolate scheme to use and of what order [1, 3].

Suppose we have progressed through the ith point in our mesh, t_i. In order to take the next step, we must compute $f_i = f(t_i, y(t_i), y(u(t_i)))$. The computation requiring a value for $y(u(t_i))$ and since $u(t_i)$ will in general be a non-mesh point, some approximation must be founded. We first find a positive integer j such that $t_{j-1} < u(t_i) \le t_j$. Since we are dealing with a retarded problem, $j \le i$. Thus sufficient past data is available to construct an approximate interpolation to y, which is then evaluated at $u(t_i)$ and this value is used to help approximating f_i. It is possible with interpolation schemes using both function and derivative values that an extrapolation must be done. This occurs if $j = i$, since in that case, f_i is not known and hence the last full set of data is available only at t_{i-1}. Such a situation was infrequent but nevertheless did occur.

Once, j is found we use the post data at mesh points t_{j-3}, t_{j-2}, t_{j-1} and t_j for a four-point interpolation scheme and for a three point interpolation scheme the data at t_{j-2}, t_{j-1} and t_j. Since the problems were solved using a variety of step sizes h the interpolation routines had to be written to accommodate this scheme.

Numerical Examples:

Example (1)

Our first equation:

$y'(t) = -y(t-1),\ t > 0$

$y(0) = 1$

$y(t) = 0,\ -1 < t < 0$

Results given for $t \in [0, 9]$.

Adaptive Runge-Kutta with Lagrange interpolation.

ε	2nd order GE	2nd order ND	4th order GE	4th order ND
10^{-3}	4.310×10^{-3}	160	1.312×10^{-3}	159
10^{-6}	4.101×10^{-5}	432	4.231×10^{-6}	429
10^{-9}	$3.810\times10^{[illegible]}$	882	$3.821\times10^{[illegible]}$	875

Adaptive Runge-Kutta with Hermite interpolation.

ε	2nd order GE	2nd order ND	4th order GE	4th order ND
10^{-3}	4.5[illegible]6$\times10^{-3}$	155	4.1667×10^{-3}	151
10^{-6}	4.561×10^{-6}	450	$4.0690\times10^{[illegible]}$	411
10^{-9}	4.123×10^{-9}	884	$3.9737\times10^{[illegible]}$	871

Note that:

ε = the required error tolerance

ND = number of derivative evaluation.

GE = maximum global error.

Example (2)

$y'(t) = -y(t-1+e^{-t}) - \sin(t-1+e^{-t}) - \cos t,\ t \ge 0$

$y(0) = \sin t,\ t \le 0$

Results for $t \in [0, 10]$ with delay $-1 - e^{-t}$

Adaptive Runge-Kutta with Lagrange interpolation.

ε	2nd order GE	2nd order ND	4th order GE	4th order ND
10^{-4}	6.0532×10^{-4}	4	$6.0031\times10^{[illegible]}$	4
10^{-6}	1.1261×10^{-6}	17	$1.046\times10^{[illegible]}$	17
10^{-10}	$4.2561\times10^{[illegible]}$	730	$4.056\times10^{[illegible]}$	720

Adaptive Runge-Kutta with Hermite interpolation.

ε	2nd order GE	2nd order ND	4th order GE	4th order ND
[illegible]	$6.0053\times10^{[illegible]}$	3	$6.0001\times10^{[illegible]}$	3
[illegible]	$1.321\times10^{[illegible]}$	6	1.031×10^{-6}	15
[illegible]	$4.123\times10^{[illegible]}$	710	$4.061\times10^{[illegible]}$	705

Step Size Control

The step size h is chosen as small as necessary to get an accurate approximation. It is chosen as big as possible to reach the end of the interval in a few steps as possible. For this reason, and to maintain the stability we can use the following adaptive procedure.

Assume that we are given a required error tolerance (ε), and that local truncation error is estimated by E, then:

(a) If $E \ge \varepsilon$, then reject the computed solution. For choosing the next step size, we have to take account of the possibility of a point of jump discontinuity in the kth derivative of solution, where $k < p+1$. Hence, we register the point $t^* = t_i + h$ as a possible point of discontinuity and then we choose the next step size as $\frac{h}{2}$. Using the step size $\frac{h}{2}$ and the solution and its derivative value at t_i we calculate the next approximation of solution and the LTE estimate E, and repeat the test in (a).

(b) If $E \le \varepsilon$, then accept the next approximation to the solution and let the next mesh point to be $t_i = t_i + h$. For choosing the next step size, we make the following tests:

If $E \ge \varepsilon/2^{p+1}$, keep the same step size and go to (iii).

If $E < \varepsilon/2^{p+1}$, then if $t^* \le t_i$ then evaluate the step size, otherwise keep the same step size and go to (iii).

If $t_i + h > T$, where T is a point where the solution is required, then we take the next step size to be $h = T - t_i$. Using the step size h and the solution and its derivative values, we calculate the next approximation and repeat the test in (a).

Conclusions

From the above results, we can notice that the fourth order Runge-Kutta method give a better results than the second order. Also, the data using Hermite interpolation is better than Lagrange.

References

1. C. A. H. Paul, "Developing a Delay Differential Equation Solver", Numerical Analysis Report, No.204, December, (1991).
2. C. A. H. Paul and C. T. H. Baker, "Explicit Runge-Kutta Methods for Numerical Solution of Singular Delay Differential Equations", Numerical Analysis Report, No.212, April, (1992).
   A. N. Al-Mutib, "Numerical Methods for Solving Delay Differential Equations", Ph.D. Thesis ,university of Manchester ,(1977).
3. L. F. Shampine and S. Thompson, "Solving Delay Differential Equations with dde23", March 23, (2000).
4. S. D. Conte and Carl de Boor, "Numerical Analysis An Algorithmic Approach", Third Edition ,McGraw-Hill book,(1980).

الخلاصة

في هذا البحث درسنا استقرارية الحلول العددية للمعادلات التفاضلية التأخيرية باستخدام طرائق رنج-كوتا الواضحة، حيث تم تقريب المعادلات التفاضلية التأخيرية باستخدام متعددات حدود لاكرانج وهيرمت للاستكمال.