Testing the Accuracy of DERIVE's "RK" Routine

Temple H. Fay and Stephan V. Joubert
Tshwane University of Technology
South Africa
joubertsv@hotmail.com

Abstract

A computer laboratory component has become part of many beginning courses on ordinary di¤erential equations. Here students are asked to numerically solve initial value problems, but little attention is given to the accuracy of such numerical solutions. Indeed, many text books only pay lip service to this important part of numerical analyses, possibly because the authors believe that the topic is too advanced for beginners. In this article we o¤er a simple DERIVE procedure that provides a measure of the accuracy of the Runge-Kutta order 4 DERIVE routine "RK". The procedure has a nice spin o¤ too: it often provides a superior numerical result from the algorithm.

1 Introduction

At the TUT in Pretoria, an introduction to ordinary di¤erential equations (ODEs) is taught in the PC laboratory, where students are asked to solve nonlinear equations and systems. Indeed, computer algebra systems, such as Mathematica [8], Maple [6], MATLAB [7], and Derive [1], have made possible a revolution in the way beginning di¤erential equations courses are being taught. The emphasis used to be on solution techniques for various classes of equations which made the courses primarily a popouri of "recipes". Today the emphasis is more upon systems, nonlinear equations, and computer explorations.

Many beginning texts discuss uniform step size numerical techniques such as Euler's method, the improved Euler's method, and many mention Runge-Kutta order 4 (without a derivation). But today's computer algebra systems generally employ a suite of much more advanced algorithms than these, for example Mathematica version 4.2 uses an Adams method of order 12 [8] to solve non-sti¤ systems.

But advanced as these algorithms are, they are by no mean infallible and even simple appearing second order equations can give rise to completely erroneous numerical solutions which appear quite plausible when plotted (for example see [3]). Thus we feel it is important for a student (or other solver) to have a method of testing the accuracy of the numerically generated solution.

One need not have a strong background in numerical analysis to understand that things can go wrong and we give a simple strategy for solving system initial value problems (IVPs) such as

$$\left.\begin{array}{ccc} \dot{x} & = & f(x,y,t) \\ \dot{y} & = & g(x,y,t) \\ x(0)=a & & y(0)=b \end{array}\right\} \tag{1}$$

(here t is the independent variable which we refer to as time and the dots refer to differentiation with respect to t) which permits testing of the accuracy of the numerical result and indeed this strategy often produces a superior numerical result from the algorithm.

2 The residual

One of the strategies suggested by Knapp and Wagon [5] to gauge how well the numerical algorithm is working and to give a measure of the accuracy of a solution is by substituting back into the equation and examining the "residual", that is to observe if the equation is "satisfied". They point out that this is not an infallible method, merely one tool in a toolbox. The strategy discussed below is a consequence of trying to compute the "residual" for a numerical solution.

We focus on second order ordinary differential equations as these are the most often encountered types in beginning courses. But there really is no restriction on the order of the equation. Suppose we wish to solve a typical second order differential equation

$$\ddot{x} + f(x, \dot{x}) = g(t), \tag{2}$$

subject to the initial conditions $x(0) = a, \dot{x}(0) = b$. The traditional approach is to introduce the auxiliary variable $y = \dot{x}$ and numerically solve the 2×2 system

$$\left.\begin{array}{ccc} \dot{x} & = & y \\ \dot{y} & = & -f(x,y) + g(t) \\ x(0)=a, \quad y(0)=b. & & \end{array}\right\} \tag{3}$$

In order to substitute back into the equation we need the second derivative of x, but this technique does not compute it. The following easy strategy produces the second derivative: Differentiate equation (2) and solve the 3×3 system

$$\left.\begin{array}{ccc} \dot{x} & = & y \\ \dot{y} & = & z \\ \dot{z} & = & -\frac{d}{dt}\{f(x,y)\} + \dot{g}(t) \\ x(0)=a, \quad y(0)=b, & & z(0) = -f(a,b) + g(0). \end{array}\right\} \tag{4}$$

By doing so, we numerically compute the second derivative of x, $z = \ddot{x}$. Mathematica does this effortlessly. It computes an "interpolating function" for $z(t)$ and so the "residual"

$$z(t) + f(x,y) - g(t)$$

can be plotted. Some more details on this may be found in [4]. The "residual method" is not always satisfactory (as is well known). To plot the "residual" in DERIVE takes quite an effort and this will not be discussed here.

3 The estimated error

We take the idea of computing a second solution using the same algorithm and step size, a little further. We shall give examples where the two solutions produced are almost identical, for a while. Using the same step size in DERIVE's RK routine these solutions can be produced using the same number of iterations and can thus easily be compared. Because the RK routine returns a matrix from which the solution $x(t)$ as well as $y(t)$ can be extracted, we can easily subtract the 2×2 system solution matrix from the 3×3 solution matrix and observe the result either row by row or graphically by appending a "time" column.

Assume that the exact solution of the IVP (1) is say $x(t)$. Assume further that the numerical solution produced by DERIVE's RK routine for the 2×2 system (3) is say $x_2(t)$ while that produced for the 3×3 system (4) is say $x_3(t)$. Define the error $E(t)$ in the RK routine to be

$$E(t) = x(t) - x_2(t) \tag{5}$$

We define the estimated error $e(t)$ in the RK routine to be

$$e(t) = x_3(t) - x_2(t) \tag{6}$$

There are many examples of IVPs with constant coefficients which are numerically unstable. In most examples we have examined, $e(t)$ is an astonishingly good estimate of $E(t)$. We discuss one such IVP (with constant coefficients) in Section 4 below. We give further evidence that $e(t)$ might be a good estimate of RK accuracy by examining an IVP with no known solution but which does have a known phase portrait in Section 5. Finally, in Section 6, we give an example of an IVP with no known solution or phase portrait, but $e(t)$ agrees with an error analyses done by Knapp and Wagon [10] using Mathematica. We believe that determining $e(t)$ is another tool which can be added to Knapp and Wagon's toolbox [5]. A similar strategy is used when solving a system (1) which cannot be written as a second order ODE. In this case, two estimated errors must be determine, one for $x(t)$ and one for $y(t)$. A snapshot of the commands used is given in the Appendix in Section 8.

4 A linear IVP

Consider

$$\left.\begin{array}{l} \ddot{x} - 2\dot{x} - 3x = -4e^{t} \\ x(0) = 2, \quad \dot{x}(0) = 0 \end{array}\right\} \tag{7}$$

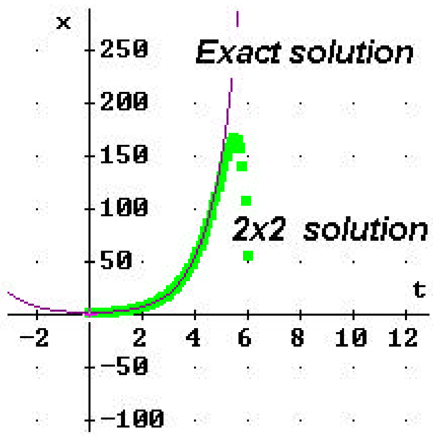

Figure 1:

This has exact solution

$$x = e^{-t} + e^{t} = 2\cosh t$$

Turning (7) into a 2×2 system

$$\left.\begin{array}{ccc} \dot{x} & = & y \\ \dot{y} & = & 2y + 3x - 4e^{t} \\ x(0) = 2, & & y(0) = 0 \end{array}\right\} \qquad (8)$$

we obtain a solution using RK with $h = 0.1$ and $n = 60$, shown in Figure 4.1. Solving the equivalent 3×3 system (4),

$$\left.\begin{array}{ccc} \dot{x} & = & y \\ \dot{y} & = & z \\ \dot{z} & = & 2z + 3y - 4e^{t} \\ x(0) = 2, & y(0) = 0, & z(0) = 2 \end{array}\right\} \qquad (9)$$

at $h = 0.1$ and $n = 60$ produces a solution which stays close to the true solution for a longer period, as shown in Figure 4.2. A close scrutiny of Figure 4.2 reveals that the two solutions start to diverge at about $t = 3$ time periods.

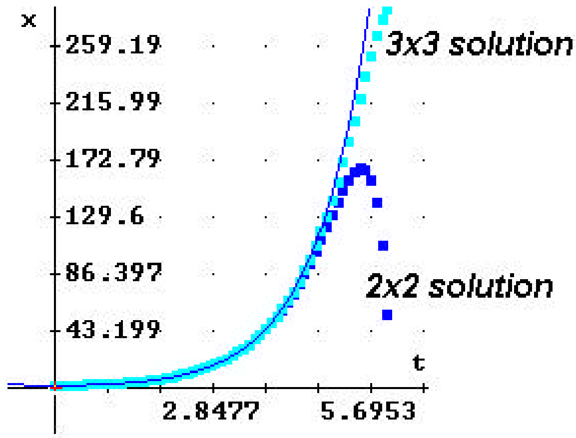

Figure 4.2.

A glance at the error tables given in Figure 4.3 shows that, we can expect three decimal accuracy for about 1.5 time periods and two decimal accuracy for 2 time periods for the RK routine with step size h = 0.1.

t	Error
1	$9.061932091 \cdot 10^{-5}$
1.1	0.0001261640890
1.2	0.0001745509993
1.3	0.0002403125765
1.4	0.0003295717135
1.5	0.0004505984321
1.6	0.0006145616139
1.7	0.0008365435693
1.8	0.001136909713
1.9	0.001543157466
2	0.002092412036

t	Estimated Error
1	$-5.943768169 \cdot 10^{-5}$
1.1	$-8.297205588 \cdot 10^{-5}$
1.2	-0.0001150529184
1.3	-0.0001587005486
1.4	-0.0002179952873
1.5	-0.0002984487011
1.6	-0.0004075046692
1.7	-0.0005552157854
1.8	-0.0007551564600
1.9	-0.001025655455
2	-0.001391459837

Figure 4.3

5 A second order IVP with know phase portrait

An innocuous appearing di¤erential equation (10), due to John Polking of Rice University, discussed brie‡y in [5] and in some detail in [9], is a numerical nightmare and hence useful for student investigations because just about everything which can go wrong does.

$$\ddot{x} - (\dot{x})^2 + x = 0 \tag{10}$$

Turning this equation into a 2×2 system

$$\left.\begin{aligned} \dot{x} &= y \\ \dot{y} &= y^2 - x \end{aligned}\right\} \tag{11}$$

it is easy to see that the origin in the phase plane is the only critical value and it is a center. There is a parabolic separatrix which divides the phase plane into two regions, one for closed bounded trajectories, the other for unbounded trajectories. But this is not at all obvious. By eliminating t and solving

$$\frac{dy}{dx} = \frac{y^2 - x}{y},$$

it turns out that the "solution" $y(x)$ is given by

$$ke^{2x} = 2y^2 - 2x - 1$$

Hence, for each initial conditions of the form $(x(0), y(0)) = (x_0, 0)$, the trajectory in the phase plane is given by the equation

$$(2x_0 + 1)e^{2(x - x_0)} - 2x - 1 + 2y^2 = 0. \tag{12}$$

A phase portrait is shown in Figure 5.1.

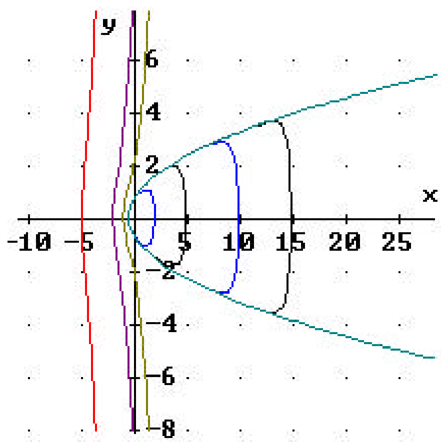

Figure 5.1.

The initial conditions $x(0) = -1/2, y(0) = 0$ determine the parabolic separatrix $x = y^2 - 1/2$. Solving the system (11) for these initial values with a step size of $h = 0.1$ with $n = 200$ we obtain the trajectory shown in Figure 5.2.

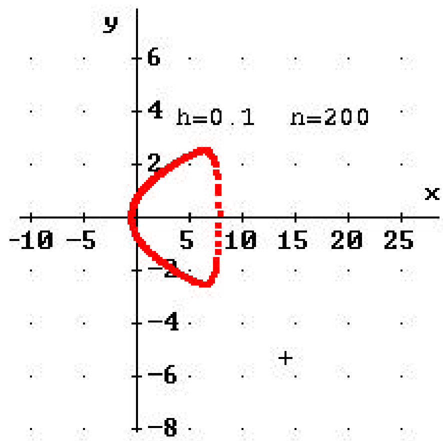

Figure 5.2.

Since we know the trajectory should be a parabola, this trajectory is clearly incorrect. Even decreasing the step size to $h = 0.05$ fails to produce the correct trajectory, it just takes longer to go off the mark, this is shown in Figure 5.3.

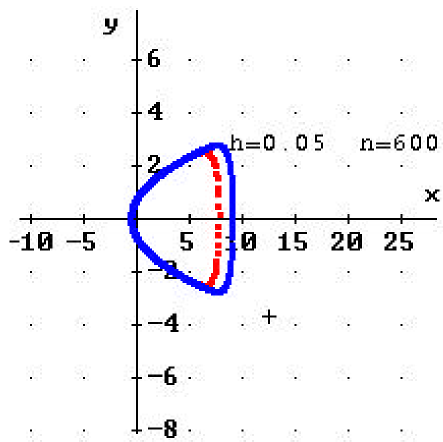

Figure 5.3.

Solving the equivalent 3×3 system (4),

$$\left.\begin{aligned} \dot{x} &= y \\ \dot{y} &= z \\ \dot{z} &= 2yz - y \\ x(0) = -1/2, &\quad y(0) = 0, \quad z(0) = 1/2 \end{aligned}\right\} \qquad (13)$$

at h = 0.1 and n = 200 produces the the correct trajectory shown in Figure 5.4.

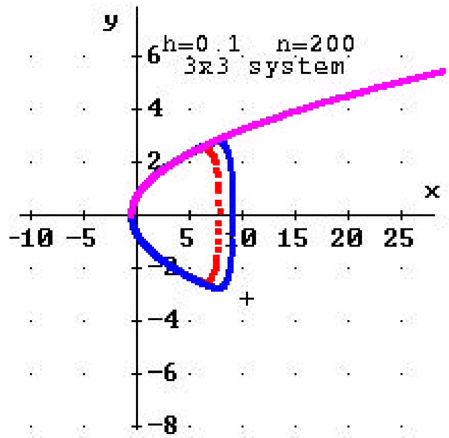

Figure 5.4.

A plot of the two solutions on the same set of axes shows that they clearly diverge from one another and this should also indicate to the student (or solver) that the algorithm is only working properly for a short time, as shown in Figure 5.5.

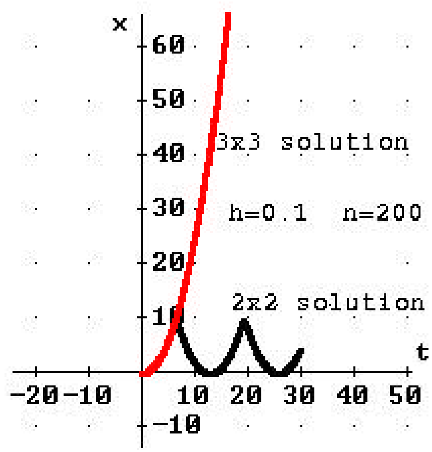

Figure 5.5.

By subtracting the two solutions we can plot the estimated error in the solution as follows in Figure 5.6.

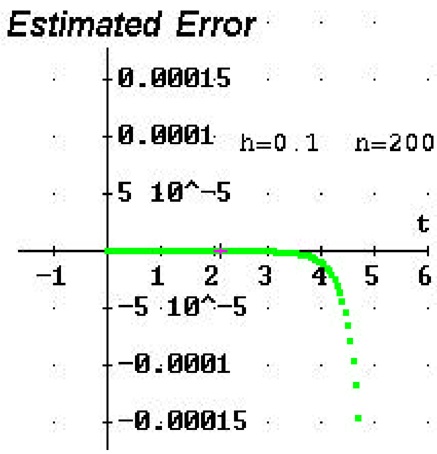

Figure 5.6.

We claim that, for this IVP, the RK routine with step size $h = 0.1$ is accurate to four decimal places for $t = 4$..

6 Du¢ng's equation

In [10] the following form of Du¢ng's equation was studied:

$$\left.\begin{aligned} &\ddot{x} + \tfrac{15}{100}\dot{x} - x + x^3 = \tfrac{3}{10}\cos(t) \\ &x(0) = -1, \quad \dot{x}(0) = 1 \end{aligned}\right\} \tag{14}$$

From Table 1 (generate by Mathematica) they predict that the solution is correct to 4 decimal places for $t = 20$ at default working precision, that is when local error is of the order 10^{-6}. We will see that, using a step size of $h = 0.032$ (that is local error is of the order $(0.032)^4 \approx 10^{-6}$) in the RK routine con…rms this. Turning this equation into a 2×2 system IVP yields:

$$\left.\begin{aligned} \dot{x} &= y \\ \dot{y} &= x - x^3 - \tfrac{15}{100}y + \tfrac{3}{10}\cos(t) \\ x(0) &= -1, \quad y(0) = 1 \end{aligned}\right\} \tag{15}$$

and a 3×3 system IVP yields:

$$\left.\begin{aligned} \dot{x} &= y \\ \dot{y} &= z \\ \dot{z} &= y - 3x^2y - \tfrac{15}{100}z - \tfrac{3}{10}\sin t \\ x(0) &= -1, \quad y(0) = 1, \quad z(0) = \tfrac{3}{20} \end{aligned}\right\} \tag{16}$$

Plots of the solutions appear in Figure 6.1.

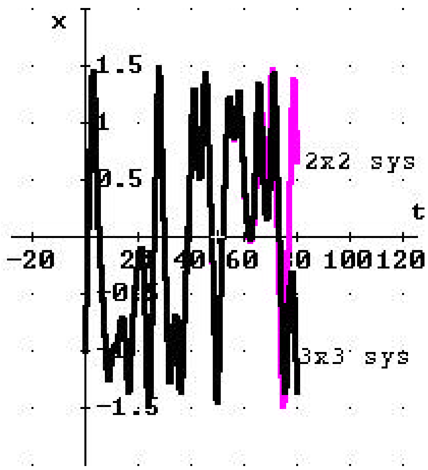

Figure 6.1.

Clearly the two solutions diverge at about t = 60, and so the RK routine is producing numerical garbage after t = 60. We estimate the error as explained above in (6) and obtain Figure 6.2, con…rming this observation.

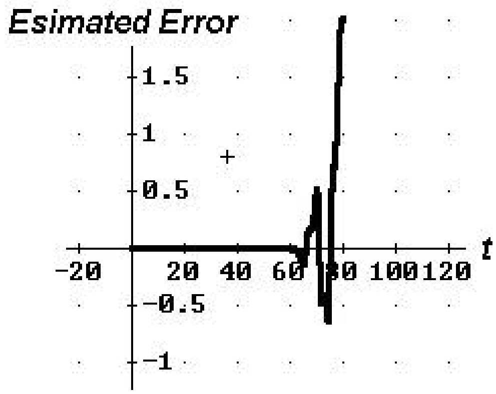

Figure 6.2.

A closer look at the graph of Estimated Error is revealed in Figure 6.3.

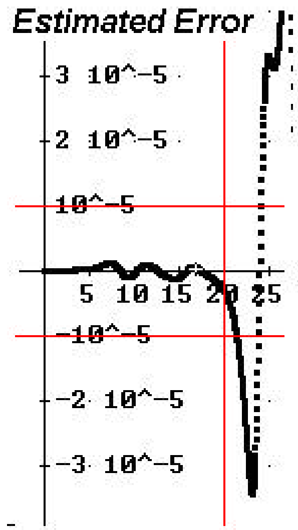

Figure 6.3.

In the region of t = 20 we see that we have four decimal place accuracy which agrees with Table 1 of [10]. Mimicking their words "we are willing to bet" that our estimated accuracy is correct for a time interval t = 20

7 Conclusion

We have selected three representative examples for which numerical di¢ culties can be observed, but there are many others. Without a priori knowledge, poor numerical performance may be di¢ cult to spot. Working with two di¤erent solutions produced by the RK routine (with the same step size) we have produced a measure of accuracy for the algorithm. Comparing these two solutions graphically may often indicate where di¢ culties arise. There are many innocuous appearing nonlinear equations that students will encounter, and it is therefore important for them to be aware that numerical problems are common, and to know how to detect them. Our technique is but one tool in a toolbox for detecting numerical problems.

References

[1] Derive Version 6, 2004, www.derive-europe.com.

[2] Fay, T. H. and J. C. Gree¤, 1999 "Lions and Wildebeest: a predator-prey model", Mathematics and Computer Education 33, 106 - 119.

[3] Fay, T. H. and S. V. Joubert, 2000, “Square waves from a black box”, Mathematics Magazine 73, 393 - 396.

[4] Fay, T. H. and S. V. Joubert, 2003, "Postanalysis of numerical solutions to ODEs", New Zealand Journal of Mathematics 32 (Supplementary Issue, November), 67 - 75.

[5] Knapp, R., and S. Wagon, 2001,"Check your answers...But how?", MATHEMATICA IN ACTION for Issue 7-4 of Mathematica in Education and Research.

[6] Maple Version 9.5, 2004, www.maplesoft.com.

[7] MATLAB Version 7, 2004, www.mathworks.com.

[8] Wolfram, S., 1996, The Mathematica Book, 3rd ed., Wolfram Media /Cambridge University Press, New York.

[9] Schwalbe, D., and S. Wagon, 1997, VisualDSolve: Visualizing Di¤erential Equations with Mathematica, Springer/TELOS, New York.

[10] Knapp, R., and S. Wagon, 1996, "Orbits worth betting on!", C²ODE²E Newsletter (Consortium for Ordinary Di¤erential Equations Experiments), Winter, 8 –13.

8 Appendix

```
#1:   CaseMode := Sensitive
#2:   InputMode := Word
```

If known, type in the exact for x(t), say x=F(t).

```
#3:   F(t) :=
```

Enter the appropriate right hand sides of the 2x2 sytem and 3x3 sytems, the initial conditions, the number of iterations, the step size and the two RK routines. Ddtg = d/dt g(x,y,t).

```
#4:   f(x, y, t) :=
#5:   g(x, y, t) :=
#6:   x0 :=
#7:   y0 :=
#8:   n :=
#9:   h :=
#10:  Sol2x2 := EXTRACT_2_COLUMNS(RK([f(x, y, t), g(x, y, t)], [t, x, y],
        [0, x0, y0], h, n), 1, 2)
#11:  Ddtg :=
#12:  z0 :=
#13:  Sol3x3 := EXTRACT_2_COLUMNS(RK([y, z, Ddtg], [t, x, y, z], [0, x0,
        y0, z0], h, n), 1, 2)
```

Iterate the exact solution (Sol),if it is known,and obtain the error ERR for the given step size h.

```
#14:  Sol := ITERATES([t + h, F(t + h)], [t, X], [0, x0], n)
#15:  ERR := EXTRACT_2_COLUMNS(APPEND_COLUMNS(TIME, Sol - Sol2x2), 1, 3)
```

Obtain the Estimated Error in the RK routine for the given step size h.

```
#16:  TIME := ITERATES([t + h], [t], [0], n)
#17:  ESTIMATED_ERR := EXTRACT_2_COLUMNS(APPEND_COLUMNS(TIME, Sol2x2 -
        Sol3x3), 1, 3)
```