Available online at http://scik.org
J. Math. Comput. Sci. 2 (2012), No. 2, 413-424
ISSN: 1927-5307

GLOBAL ERROR CONTROL IN A LOTKA-VOLTERRA SYSTEM USING THE RKQ ALGORITHM

J.S.C. PRENTICE *

Department of Applied Mathematics, University of Johannesburg, South Africa

Abstract. We use explicit, implicit and symplectic versions of Euler's method to solve a Lotka-Volterra system. We implement local error control via local extrapolation, and we use the RKQ algorithm, which provides global error control. We find that local extrapolation, although providing acceptable control of local error, does not control global error. However, RKQ achieves both local and global error control in a stepwise manner. A symplectic form of Euler's method preserves the first integral of the system if the stepsize is fixed, but not if an explicit method is used in the local extrapolation mode. However, the explicit form of RKQ does preserve the first integral, because it is able to control global error.

Keywords: RKQ, RKrvQz, Lotka-Volterra, Global error, Error control, Runge-Kutta, Quenching.

2000 AMS Subject Classification: 65L05, 65L06, 65L70

1. Introduction

Lotka-Volterra systems are used to model predator-prey type populations, where the populations of two rival species rise and fall in relation to each other. These systems are often used as examples in discussions of numerical methods for initial-value problems [1,

*Corresponding author

E-mail address: jprentice@uj.ac.za (J.S.C. Prentice)

Received January 23, 2012

2]. In this paper, we consider

$$u' = u\,(v-1)\,, \quad v' = v(2-u) \tag{1}$$

as a typical case, and seek the solution using variants of Euler's method, as in [1]. It should be noted that (1) has a periodic solution, so that it is not necessary to integrate over more than a few periods to understand the solution. However, (1) has the more general form

$$u' = Auv + Bu, \quad v' = Cuv + Dv$$

where the coefficients A, B, C and D may be time dependent. For example, they may represent seasonal parameters of the ecosystem in which the populations u and v exist. Consequently, it is appropriate to actually consider the long-time integration of such a system. We will not engage in population modelling here; rather, we will use the periodic system (1) as a simulation of a particular feature that we wish to study. This feature is the accumulation of global error over long intervals of integration, and how we could remedy such a problem by means of the RKQ algorithm.

Furthermore, we will restrict our work to results obtained using Euler's method, since this is the method used in [1], but we will make mention of the use of higher-order methods.

2. Preliminaries

We refer to an *explicit* Runge-Kutta method of order r as RKr. Implicit and symplectic methods carry a prefix of I or S.

2.1 Euler's methods

We use the explicit Euler method (RK1)

$$y_{n+1} = y_n + hf\,(t_n, y_n)$$

which gives, for (1),

$$\begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} u_n \\ v_n \end{bmatrix} + h \begin{bmatrix} u_n\,(v_n - 1) \\ v_n\,(2 - u_n) \end{bmatrix},$$

and the implicit Euler method (IRK1)

$$y_{n+1} = y_n + hf(t_n, y_{n+1})$$

which gives

$$\begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} u_n \\ v_n \end{bmatrix} + h \begin{bmatrix} u_{n+1}(v_{n+1} - 1) \\ v_{n+1}(2 - u_{n+1}) \end{bmatrix}.$$

This is a 2×2 nonlinear system which can be solved using Newton's method or some variant thereof [3].

We will also consider the symplectic Euler method (SRK1) for this problem

$$\begin{aligned} \begin{bmatrix} u_{n+1} \\ v_{n+1} \end{bmatrix} &= \begin{bmatrix} u_n \\ v_n \end{bmatrix} + h \begin{bmatrix} u_n(v_{n+1} - 1) \\ v_{n+1}(2 - u_n) \end{bmatrix} \\ &= \begin{bmatrix} \frac{hu_nv_n}{hu_n - 2h + 1} - hu_n + u_n \\ \frac{v_n}{hu_n - 2h + 1} \end{bmatrix}. \end{aligned}$$

2.2 The RKrvQz algorithm

In this algorithm, we apply local extrapolation using two RK methods, RKr and RKv $(r < v)$ to control local error (we will use the symbol LE(RKr,RKv) to denote this process), but we also apply local extrapolation with RKr and RKz $(r, v \ll z)$. This enables us to determine both local error in the RKr solution, and the global error that is propagated from the previous node and, hence, the total global error at the node of interest. If this global error exceeds the desired tolerance, we *quench* the RKr and RKv solutions – this simply involves replacing them with the RKz solution, which is assumed to be much more accurate. We then proceed to the next node. This algorithm can provide *in situ* control of the global error – the global error is kept within the desired tolerance as the RK computation proceeds - provided that the RKz solution is suitably accurate. The reader is referred to our previous work for a more detailed discussion of RKQ [4, 5, 6].

3. Calculations

We solve (1) with the initial condition

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 2.725 \\ 1 \end{bmatrix}$$

as in [1]; we define $t_0 \equiv 0$ and integrate up to $t = 100$. The solution is periodic with period $T \sim 4.5$. The trajectory (u, v) forms a closed curve in the first quadrant in phase space. Broadly speaking, u oscillates between 1.4 and 2.8, and v oscillates between 0.6 and 1.54. Since these solutions are of order unity, we apply absolute error control only (as opposed to relative error control).

We use LE(RK1,RK2) and RK12Q8, where RK2 is the second-order (explicit) Trapezium method, and RK8 is due to Fehlberg [7, 8]. In other words, we find a solution with local error control only, and then a solution where global error has been controlled via RKQ. We impose a tolerance of 10^{-2} on both local and global error. In Figures 1 and 2 we show the local error and global error for LE(RK1,RK2) and the global error for RK12Q8.

Next, we consider LE(IRK1,RK2) and IRK12Q8, also with a tolerance of $\delta = 10^{-2}$. Note that we use the implicit Euler and the explicit RK2 in this calculation. Errors are shown in Figures 3 and 4.

We also consider the use of SRK1. Since (1) possesses the first integral

$$I(u, v) = 2\ln u - u + \ln v + v$$

we expect that SRK1 will preserve $I(u, v)$. We will show that LE(SRK1,RK2) does not preserve $I(u, v)$ (although SRK1 does), but that SRK12Q8 does, via global error control.

3. Discussion

We see in Figures 1 and 2 that the local error in each component is bounded over the interval of integration; indeed, its magnitude never exceeds 0.0099 for either component, clearly less than the tolerance of $\delta = 10^{-2}$. On the other hand, the global error exhibits an increasing trend, attaining maximum magnitudes of 0.125 for v, and 0.182 for u. Obviously, controlling the local error has not resulted in an acceptably small global error. Also in these figures we see that the global error in each component, using RK12Q8, is clearly bounded - maximum magnitudes are 0.0099 and 0.0089 for u and v, respectively. RK12Q8 has certainly kept the global errors within the desired tolerance. We remind the reader that RK12Q8 also achieves local error control, in addition to global error control.

Similar results are obtained for LE(IRK1,RK2) and IRK12Q8 (see Figures 3 and 4). For LE(IRK1,RK2) the maximal local errors are 0.01 and 0.008 in u and v, and the maximal global errors are 0.27 and 0.18 in u and v, respectively. For IRK12Q8, the maximal global errors are 0.0098 and 0.0086 in u and v, respectively. Again, the RKQ algorithm has been successful.

In Figure 5 we plot $I(u,v)$ for three cases: LE(SRK1,RK2), SRK12Q8 and SRK1. The SRK1 computation has a fixed stepsize ($h = 0.1$) with no error control at all. We find that $I(u,v)$ oscillates about its expected value of -1.72006 for both SRK1 and SRK12Q8, with the latter showing oscillations of smaller amplitude. On the other hand, the LE(SRK1,RK2) solution shows a general drift in $I(u,v)$ - we believe this is due to the use of the explicit (nonsymplectic) method RK2 as part of the error control device, which serves to destroy the symplecticity of the SRK1 solution. Nevertheless, it is, in fact, the RKQ algorithm that provides the best result here - for this solution the first integral is qualitatively similar to that for SRK1, but with smaller variation. The mean value of $I(u,v)$ for SRK1 is -1.718, while the mean value for SRK1Q8 is -1.7202. We can obtain some understanding of why RKQ would be capable of reproducing a 'well-behaved' first integral: assuming global errors Δu and Δv in u and v, we find

$$\begin{aligned} I(u+\Delta u, v+\Delta v) &= 2\ln(u+\Delta u) - u + \Delta u + \ln(v+\Delta v) + v + \Delta v \\ &\approx 2\ln u - u + \ln v + v \\ &\quad + \left(\frac{\Delta u}{u} - \Delta u + \frac{\Delta v}{v} + \Delta v\right) \end{aligned} \tag{2}$$

after appropriate Taylor expansions and ignoring higher-order terms. The term in parentheses in (2) is an error term and is clearly proportional to the global errors Δu and Δv. Given the ranges of u and v we have

$$\left|\frac{\Delta u}{u} - \Delta u + \frac{\Delta v}{v} + \Delta v\right| \lesssim \left|\Delta u\left(\frac{1}{1.4} + 1\right)\right| + \left|\Delta v\left(\frac{1}{0.6} + 1\right)\right| \leqslant 4.4\delta.$$

Hence, if δ is small, the oscillations in $I(u, v)$ will also be small.

Lastly, in Figure 6 we show the estimated global error in the RK8 solution. Clearly, since the error is so small, the RK8 solution is a good estimator of global error in the Euler solutions. This error was estimated using the relationship between local and global errors

$$\Delta_{n+1} = \varepsilon_{n+1} + \alpha_n \Delta_n.$$

The reader is referred to [4, 8] for further information regarding this equation, including the definition of symbols.

Since our objective here has been simply to illustrate the nature of the RKQ algorithm, with respect to Euler's method, we have chosen a moderate tolerance of 10^{-2}. A stricter tolerance could have been imposed; this would simply have led to smaller stepsizes but the essential results would have been unchanged. We would still have found RKQ to provide good global error control, while LE would not have done so. This said, strict tolerances are better implemented using higher-order methods, which are more efficient. Such algorithms would be RK34Q8 or RK45Q8, for example (in the latter, the RK45 component could be Fehlberg's embedded method [9, 10]). In fact, we can state that we have solved (1) using RK45Q8 subject to $\delta = 10^{-8}$. A good example of the use of RK34Q8 has been given in [11].

3. Conclusion

We have used variants of Euler's method to solve a Lotka-Volterra system. We have obtained solutions by imposing local error control (via local extrapolation) only, and by imposing both local and global error control (via the RKQ algorithm). Our results clearly show that local extrapolation, although providing acceptable control of local error, does not control global error. By contrast, RKQ achieves both local and global error control in a stepwise manner (as the Runge-Kutta integration proceeds). Solutions obtained using a symplectic form of Euler's method preserve the first integral if the stepsize is fixed, but do not if an explicit method is used in the local extrapolation mode. However, the explicit form of RKQ does preserve the first integral, due to its control of global error. This work

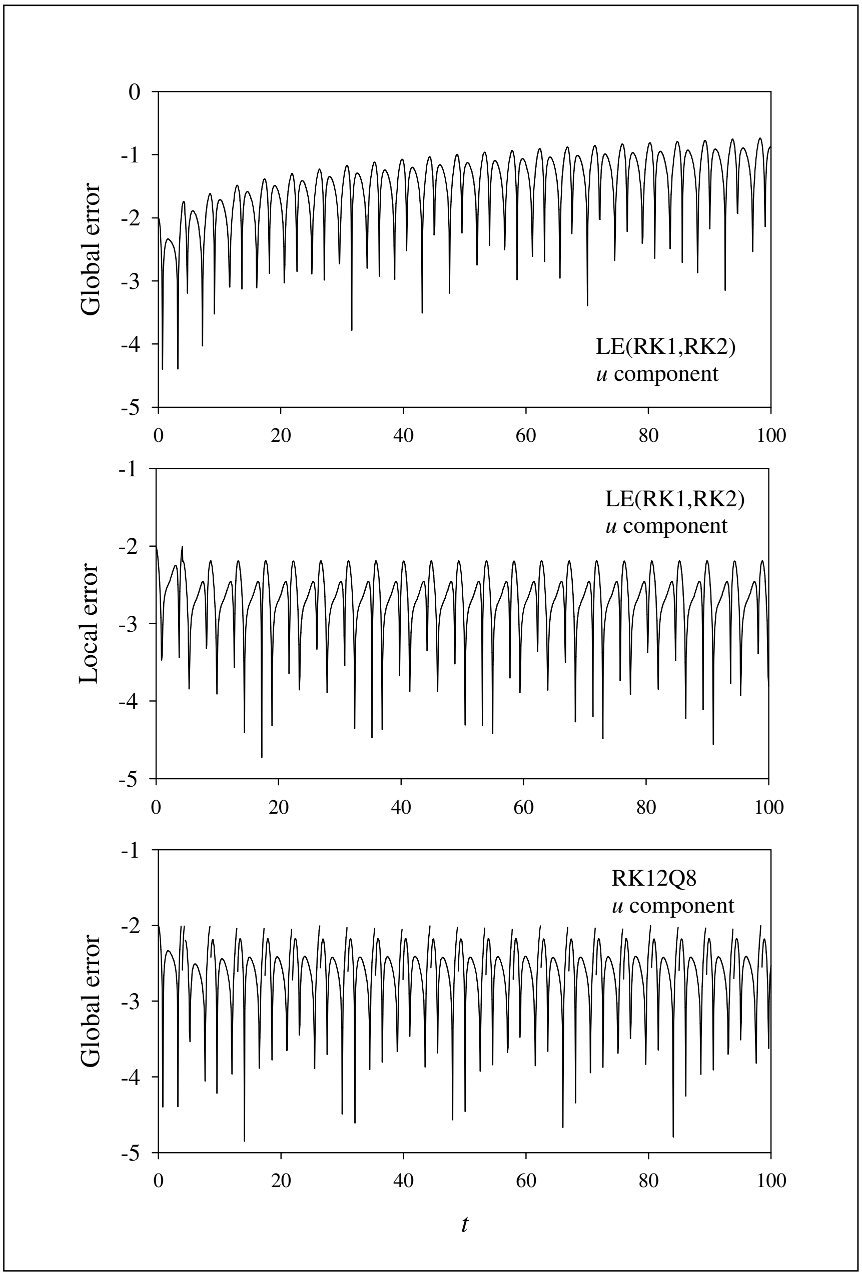

FIGURE 1. Global error (top), local error (middle) and global error (bottom) in the u component of the Lotka-Volterra system, for the indicated algorithms. Vertical axes show base 10 exponents.

clearly demonstrates the virtues of controlling global error as the integration proceeds, rather than merely controlling local error.

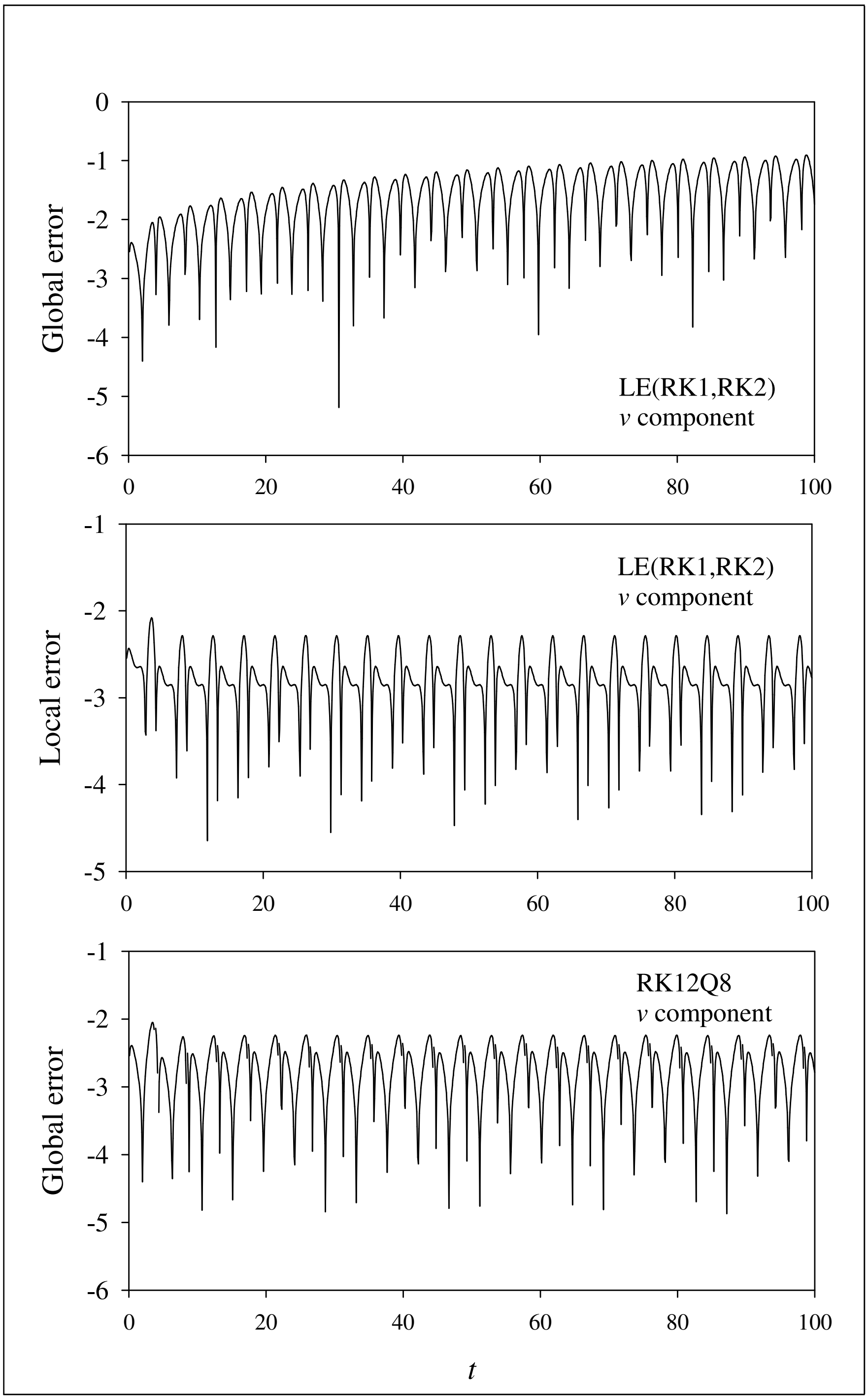

FIGURE 2. Global error (top), local error (middle) and global error (bottom) in the v component of the Lotka-Volterra system, for the indicated algorithms. Vertical axes show base 10 exponents.

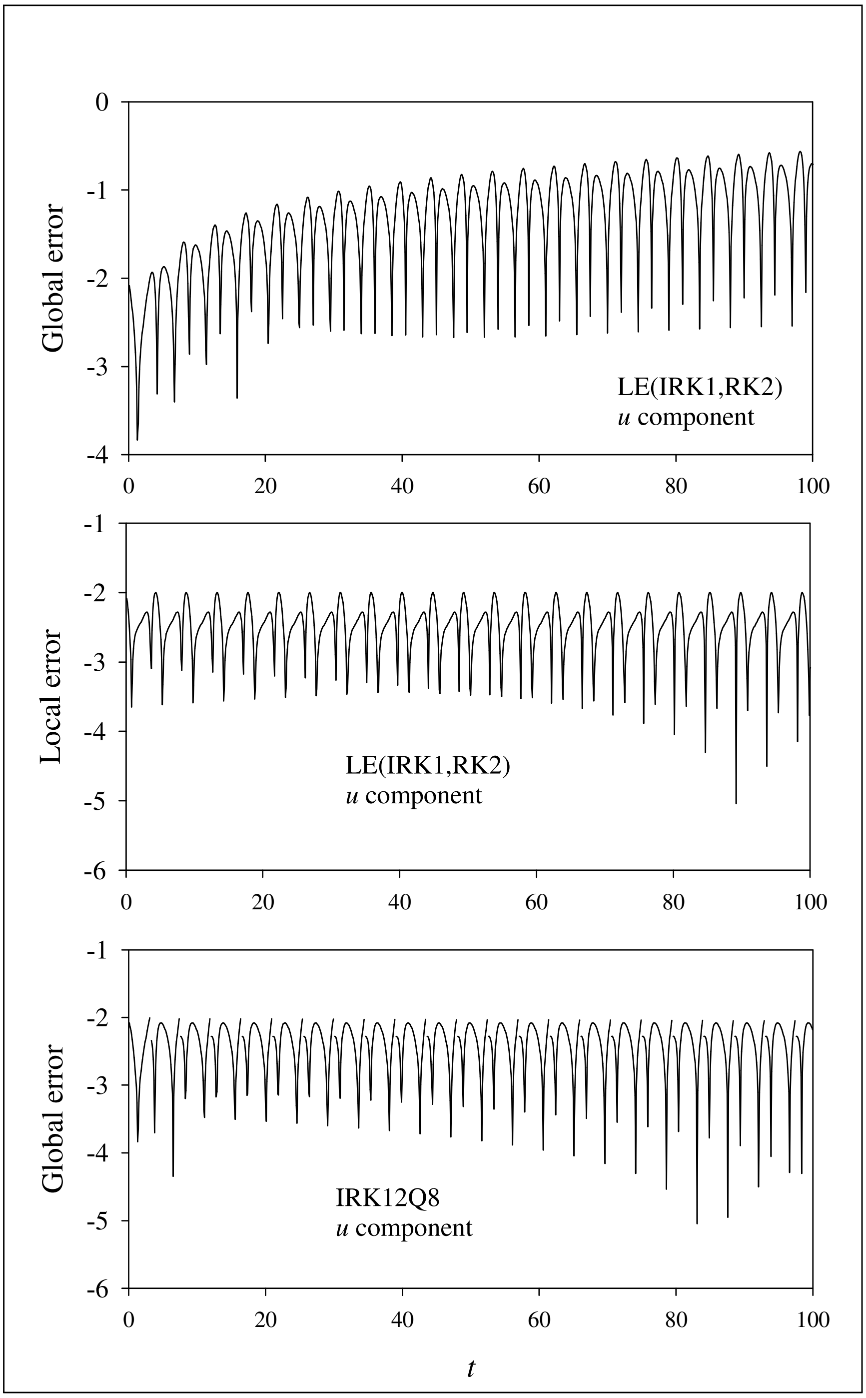

FIGURE 3. Global error (top), local error (middle) and global error (bottom) in the u component of the Lotka-Volterra system, for the indicated algorithms. Vertical axes show base 10 exponents.

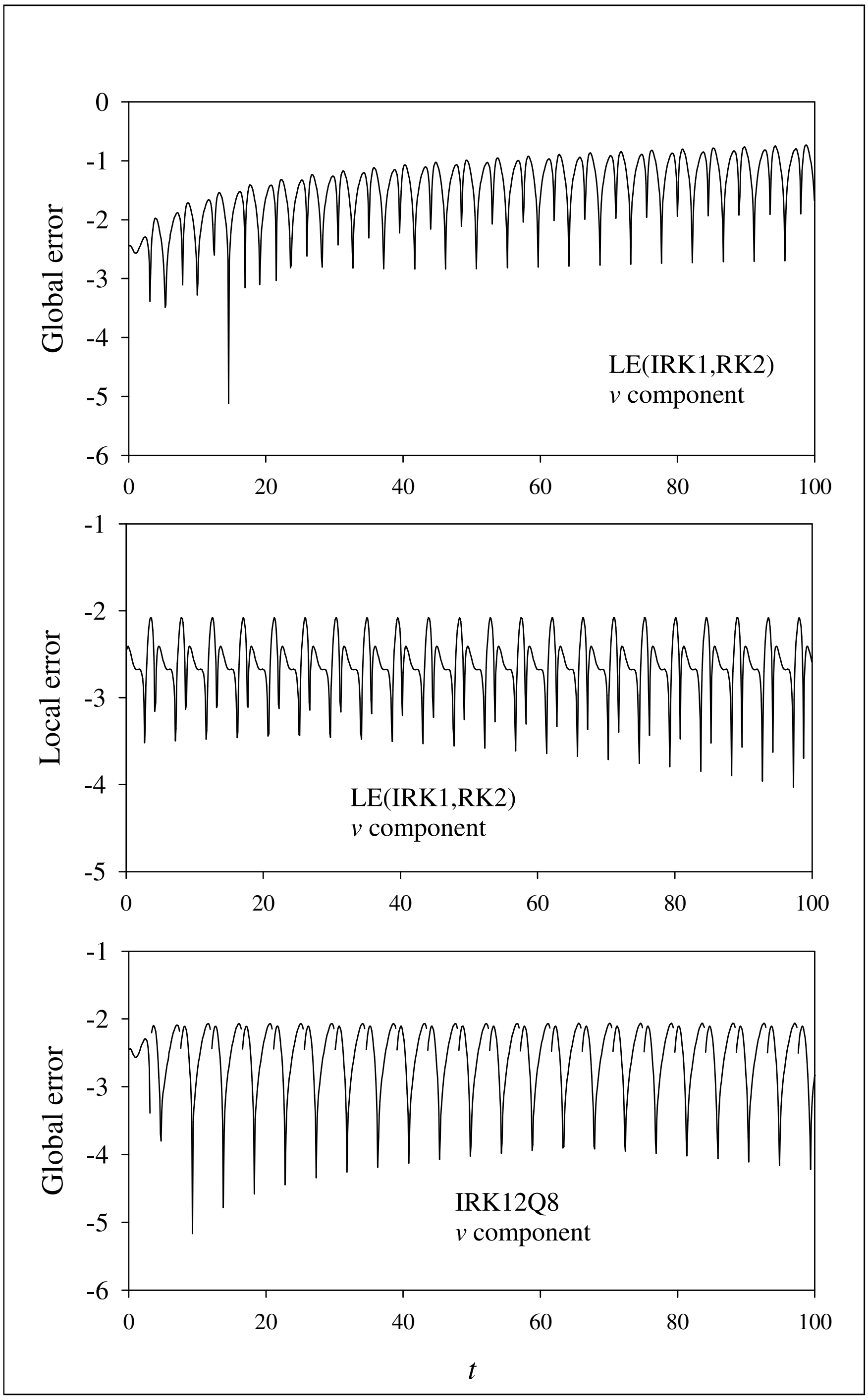

FIGURE 4. Global error (top), local error (middle) and global error (bottom) in the v component of the Lotka-Volterra system, for the indicated algorithms. Vertical axes show base 10 exponents.

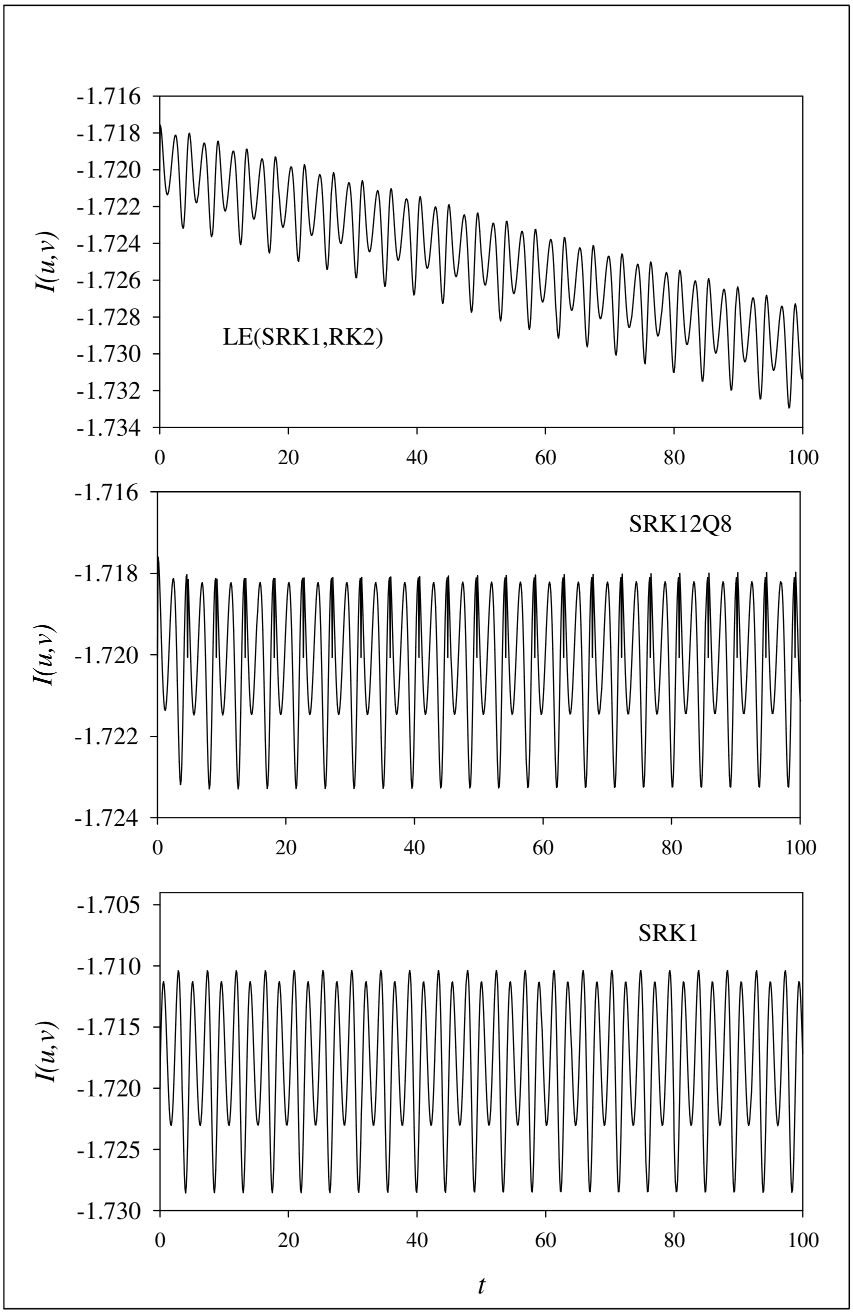

FIGURE 5. First integral $I(u, v)$ for the indicated algorithms. The drift for LE(SRK1,RK2) is obvious.

REFERENCES

[1] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II,* Berlin: Springer, 2002.

[2] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations,* Chichester: Wiley, 2003.

[3] J.S.C. Prentice, First-principles derivation of a third-order method for solving a two-dimensional nonlinear system, *Journal of Mathematics Research*, 2, 4 (2010) 57-63

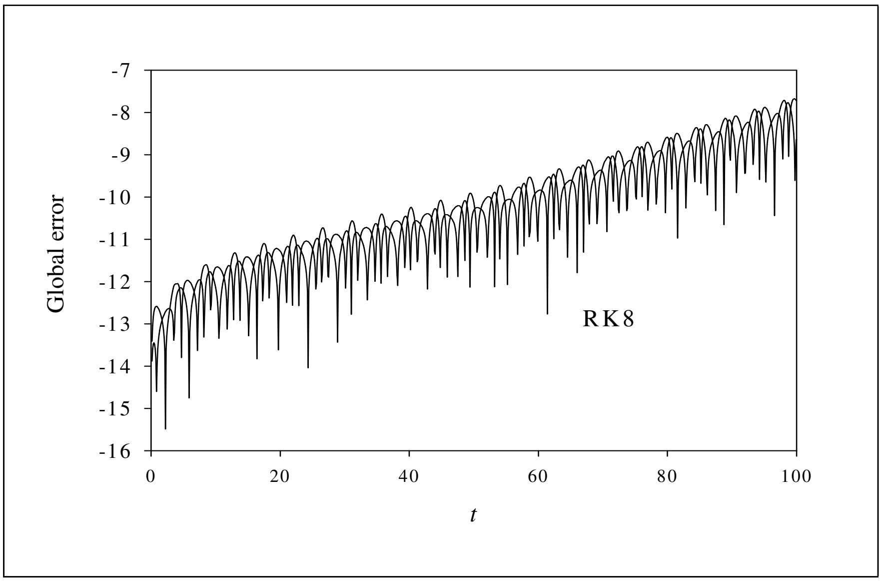

FIGURE 6. Estimated global error for RK8 for both components of the Lotka-Volterra system. Vertical axis shows base 10 exponents.

[4] J.S.C. Prentice, Stepwise Global Error Control in an Explicit Runge-Kutta Method using Local Extrapolation with High-Order Selective Quenching, *Journal of Mathematics Research,* 3, 2 (2011) 126-136.

[5] J.S.C. Prentice, Relative Global Error Control in the RKQ Algorithm for Systems of Ordinary Differential Equations, *Journal of Mathematics Research,* 3, 4 (2011) 59-66.

[6] J.S.C. Prentice,. Nyström Methods in the RKQ Algorithm for Initial-value Problems, *arXiv.org* (2011), [arXiv:1110.6749]

[7] E. Fehlberg, Classical fifth-, sixth-, seventh-, and eighth order Runge-Kutta formulas with step size control, *NASA Technical Report 287* (1968); extract published in *Computing*, 4 (1969) 93-106.

[8] E. Hairer, S.P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Berlin: Springer, 2000.

[9] J.S.C. Prentice,. General error propagation in the RKrGLm method, *Journal of Computational and Applied Mathematics,* 228 (2009) 344-354.

[10] E. Fehlberg, Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems, *NASA Technical Report 315* (1969); extract published in *Computing*, 6 (1970) 61-70.

[11] J.S.C. Prentice, Global Error Control in the Runge-Kutta Solution of a Hamiltonian System, *arXiv.org* (2011), [arXiv:1111.6996]