Proceedings of the World Congress on Engineering 2010 Vol III
WCE 2010, June 30 - July 2, 2010, London, U.K.

An Instance of Failure for the MATLAB Explicit `ODE45` Solver

Riccardo Fazio *

Abstract—**We consider the adaptive strategies applicable to a simple model describing the phase lock of two coupled oscillators. This model has been used to show an instance of failure of the `ODE45` Runge-Kutta-Felberg solver implemented within the MATLAB ODE suite, see [J. D. Skufca. Analysis still matters: a surprising instance of failure of Runge-Kutta-Felberg ODE solvers. SIAM Review, 46:729-737, 2004]. We compare the numerical results obtained with: the MATLAB ODE suite's explicit solvers, and the local linearity strategy implemented with the classical fourth-order Runge-Kutta method as a basic method.**

Keywords: initial value problems, adaptive numerical methods, local linearity approach.

1 Introduction

We consider the adaptive strategies used for the numerical integration of initial value problems (IVPs) governed by systems of ordinary differential equations

$$\left\{ \begin{array}{ll} \dfrac{du}{dt} = \mathrm{f}(u) \ , & t \in [t_0, t_{\max}] \\ u(t_0) = u_0 \ , & \end{array} \right. \tag{1}$$

where $u(t) : \mathbb{R} \to \mathbb{R}^k$, $u_0 \in \mathbb{R}^k$ and $\mathrm{f}(u) : \mathbb{R}^k \to \mathbb{R}^k$. Accepted strategies for variable step size selection are based mainly on the inexpensive monitoring of the local truncation error, or the residual monitoring, or the definition of a suitable monitor function, or the utilization of scaling invariance properties. The relevant bibliography can be listed as follows.

1) Local error control, first introduced by Milne's device in the implementation of predictor-corrector methods [11, pp. 107-109] or [9, pp. 75-81]. Extensions to embedded Runge-Kutta methods have been developed by Sarafyan [12], Fehlberg [8], Verner [18] and Dormand and Price [4].

2) Local error control based on Richardson local extrapolation, see Shampine [15, pp. 361-364].

3) Residual (or size of the defect) monitoring, proposed by Enright [5], see also his survey paper [6].

4) Monitoring the relative change in the numerical solution, proposed by Shampine and Witt [14] and recently modified by Jannelli and Fazio [10].

5) Adaptivity by scaling invariance, proposed for the numerical solution of blow-up problems by Budd et al. [1, 2].

We report here an application of a new strategy based on monitoring the approximate local linearity of the computed solution. For this strategy, preliminary numerical results, concerning the classical two body problem, were presented at the World Congress on Engineering held in London (July 1-3, 2009) [7].

2 Two coupled oscillators

Let us consider a system modeling two coupled oscillators

$$\left\{ \begin{array}{l} \dfrac{d\theta_1}{dt} = \omega_1 + k_1 \ \sin(\theta_2 - \theta_1) \\ \dfrac{d\theta_2}{dt} = \omega_2 + k_2 \ \sin(\theta_1 - \theta_2) \ , \end{array} \right. \tag{2}$$

where θ_1 and θ_2 are the two phase angles describing the time evolution of the two oscillators, with natural frequencies ω_1 and ω_2, respectively, k_1 and k_2 are the coupling constants between the oscillators, and t is the time independent variable. The model (2) has been proposed and studied as a dynamicsl system [17]. As a specific test case, we will consider here the initial value problem

$$\left\{ \begin{array}{ll} \dfrac{d\theta_1}{dt} = 1 + \sin(\theta_2 - \theta_1) \ , & t \in [0, t_{\max}] \\ \dfrac{d\theta_2}{dt} = 1.5 + \sin(\theta_1 - \theta_2) & \\ \theta_1(0) = 3 \ , \quad \theta_2(0) = 0 \ . & \end{array} \right. \tag{3}$$

This problem has the exact solution:

$$\begin{aligned} \theta_1(t) &= \frac{5}{4}\, t + \frac{3}{2} - \arctan(p1) \\ \theta_2(t) &= \frac{5}{4}\, t + \frac{3}{2} + \arctan(p1) \end{aligned} \tag{4}$$

*Department of Mathematics, University of Messina, Contrada Papardo, Salita Sperone 31, 98166 Messina, Italy. E-mail: rfazio@dipmat.unime.it Home-page: http://mat520.unime.it/fazio
Acknowledgement. This work was supported by the University of Messina and partially by the Italian MUR. Date of the manuscript submission: January 19, 2010.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

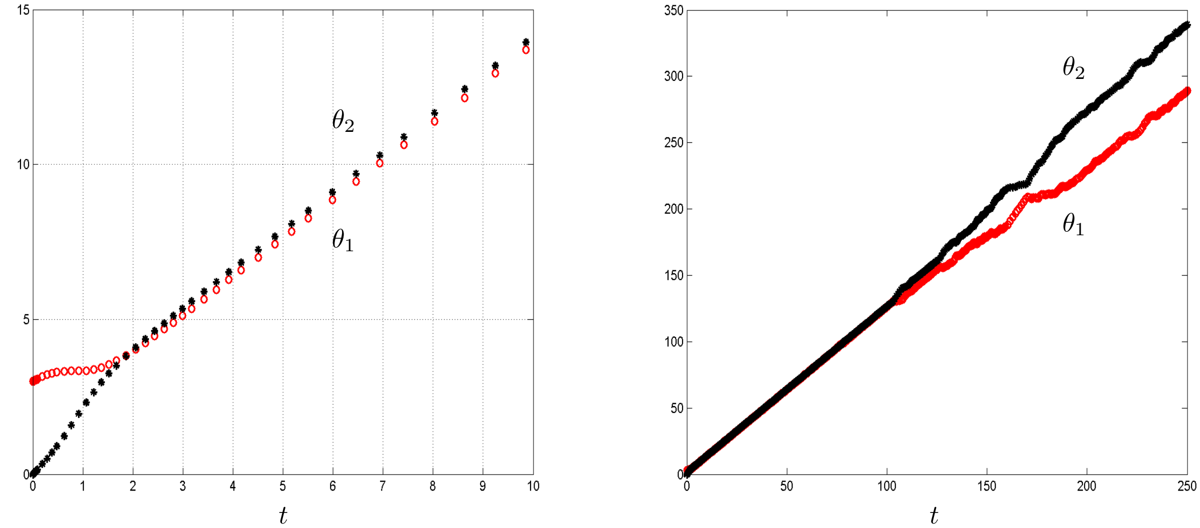

Figure 1: Numerical solution of (3) for $t \in [0, 250]$ by the MATLAB solver `ODE45`. Left: zoom of the initial stage. Right: longer computation.

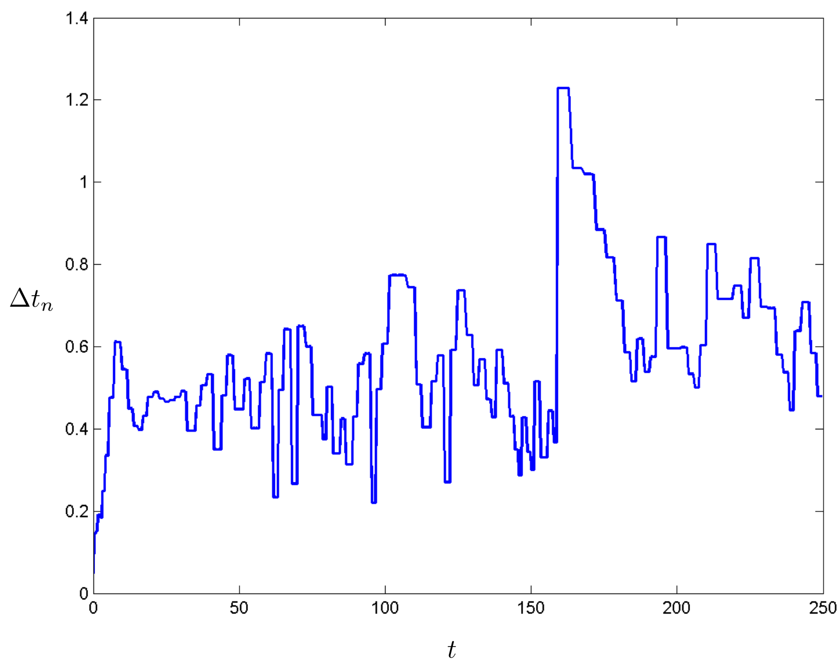

Figure 2: Step-sizes used by `ODE45` in solving (3).

where $p1 = (4\ p2 - 15\ \tanh((15\ t + 4\ p2\ \mathrm{arctanh}(p2\ (\tan(3/2) + 4)/15))\ p2/60))\ p2/15$ and $p2 = 15^{(1/2)}$.

By following Skufca [16], let us try to solve problem (3) by the `ODE45` solver with the accuracy and adaptivity parameters defined by default. Figure 1 shows the numerical results. The two oscillators phase look by about $t = 2$, but they decorrelate for larger values of t shortly after $t = 100$. It has been shown by Skufca that the two oscillators phase lock has to be preserved for all times after the correlation [16, p. 731]. Figure 2 plots the selected steps. It is worth noticing that the step-sizes

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

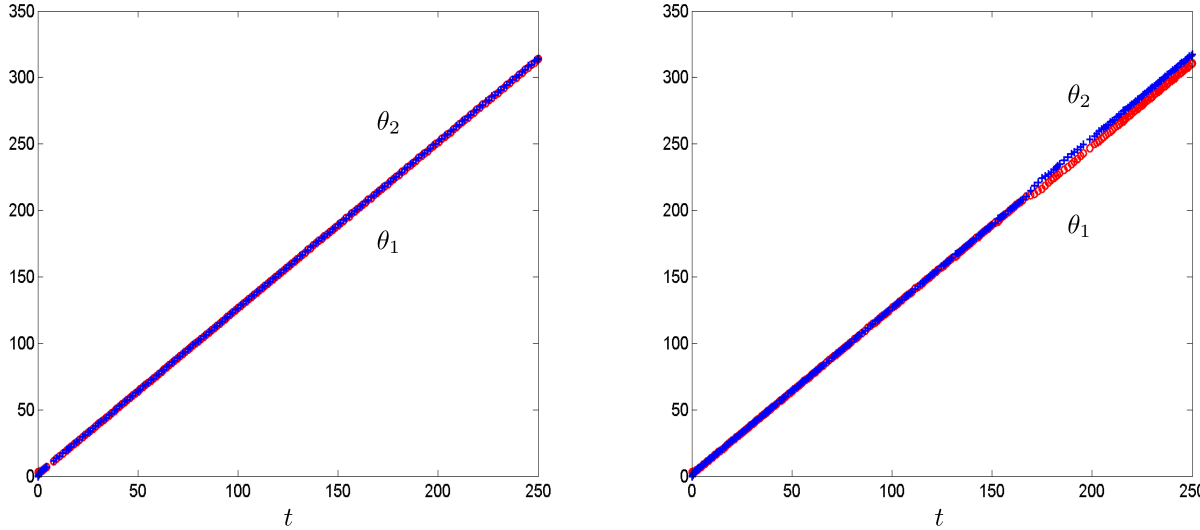

Figure 3: Numerical solution of (3) for $t \in [0, 250]$. Left: by `ODE23`. Right: by `ODE113`.

shown in figure 2 are smaller than those found by Skufca and reported on figure 2 of [16]. In particular, the adaptivity algorithm of `ODE45` has been modified in order to take care of the stability limit $\Delta t_n < 1.44$ suggested by Skufca in the analysis of section 3 of his paper. So that, it is surprising to find that the decorrelation of the two oscillators is still present in figure 1. Indeed, using the `odeset` option command, Skufca should have verified that by imposing $\Delta t_{\max} < 1.44$ he would be able to compute a correlated numerical solution, but he didn't. So that, we decided to use the classical fourth order Runge-Kutta (RK4) method, see Butcher [3, p. 166], a step-size update formula similar to the one applied by `ODE45`, and to look for a different adaptive strategy.

2.1 MATLAB explicit ODEs solvers

The numerical results given by the `ODE45` solver have been the topic of this section. For the sake of completeness, figure 3 displays the numerical results obtained by the MATLAB explicit solvers `ODE23` and `ODE113`. It is easily seen that, for the considered problem, `ODE23` is more accurate than `ODE113`.

In the next section we describe our local linearity monitoring.

3 Local linearity and adaptivity

This novel approach is based on the idea that, locally, every continuous solution behaves approximatively like a straight line. Therefore, a new monitor function can be defined as follows

$$\vartheta_n = \frac{2r}{1+r} \max_{j=1,\dots,k} \frac{\left| {}^j u_{n+1} - (1+r)\, {}^j u_n + r\, {}^j u_{n-1} \right|}{| {}^j u_n | + \epsilon} \qquad (5)$$

where $r = \Delta t_n / \Delta t_{n-1}$, $0 < \epsilon \ll 1$, and we use the notation for the components of a vector introduced by Lambert [11, p. 3], so that ${}^j u$ is the j-th component of the vector u. Note that our monitor function (5) reduces to

$$\vartheta_n^* = \max_{j=1,\dots,k} \frac{\left| {}^j u_{n+1} - {}^j u_n \right|}{| {}^j u_n | + \epsilon} \qquad (6)$$

when we set ${}^j u_{n-1} = {}^j u_n$ and $r = 1$. Therefore, at the initial step, we can apply ϑ_n^* insted of ϑ_n only by setting the two mentioned conditions.

In order to show the meaning of our control function we recall the finite difference approximation

$$\frac{d^2 u}{dt^2}(t_n) = \frac{2r}{1+r} \frac{u_{n+1} - (1+r)\, u_n + r\, u_{n-1}}{(\Delta t_n)^2} + O(\Delta t_n) \ , \qquad (7)$$

where the first two addends of the error term $O(\Delta t_n)$ are given by

$$\frac{(1-r)}{r} \frac{\Delta t_n}{3} \frac{d^3 u}{dt^3}(t_n) + \frac{(r^2 - r + 1)}{r^2} \frac{(\Delta t_n)^2}{12} \frac{d^4 u}{dt^4}(t_n) \ .$$

Therefore, our monitor function, defined by equation (5), is a first order finite difference approximation for

$$\vartheta_n \approx (\Delta t_n)^2 \max_{j=1,\dots,k} \frac{\left| \dfrac{d^2\, {}^j u}{dt^2}(t_n) \right|}{| {}^j u(t_n) | + \epsilon} \ .$$

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

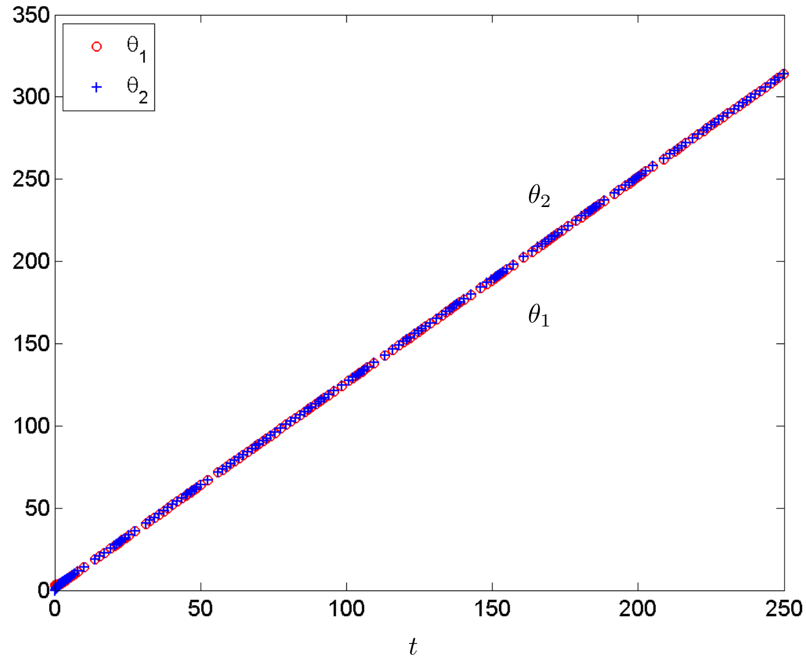

Figure 4: Local linearity control. Numerical solution for the problem 3.

We can also note that, if we set $\Delta t = \Delta t_n = \Delta t_{n-1}$, then the finite difference formula (7) reduces to the classical second order central approximation

$$\frac{d^2u}{dt^2}(t_n) = \frac{u_{n+1} - 2\,u_n + u_{n-1}}{(\Delta t)^2} + O\left[(\Delta t)^2\right] \ ,$$

where the error term $O\left[(\Delta t)^2\right]$ has only even powers of Δt.

4 Step size selection

As far as the adaptivity control is concerned, we can require that the step size selection is such that ϑ_n satisfies the condition

$$0 < \vartheta_n \le \tau \ ,$$

where τ is a user defined tolerance bound.

As far as the control of the local error estimate is concerned, it has been shown by Shampine [13] that, both in the case when the step is rejected and repeated, or in the case of a successful step, the largest step size that can be used in order to get the next step a successful one, is given by

$$\Delta t = \Delta t_n \left(\frac{\tau}{\vartheta_n}\right)^{1/(p+1)} \ , \qquad (8)$$

where p is the order of the method used. However we are considering a different control strategy, and therefore we are willing to apply two safe parameters, say $0 < s_1 < 1$ e $s_2 > 1$, for the selection of the new time step. So that, we can use the predicted new step size

$$\Delta t_{n+1} = \begin{cases} s_2\ \Delta t_n & \text{if} \quad s_1\ \Delta t > s_2\ \Delta t_n \\ \Delta t_n/s_2 & \text{if} \quad s_1\ \Delta t < \Delta t_n/s_2 \\ s_1\ \Delta t & \text{otherwise} \ . \end{cases} \qquad (9)$$

In this way, we apply a reduction factor s_1 of the predicted step size and, moreover, it will be true that the amplification, or reduction, factor of the previous step size never exceeds the value s_2.

In any case, the user have to define the following adaptive parameters: a tentative initial step size Δt_0, an upper bound tolerance τ, a safe factor s_1, and a step amplification and reduction factor s_2.

We remark that the linearity adaptive approach is based on implementing only one numerical method, and that, in order to advance the computation, it uses three numerical approximations obtained at three consecutive time steps.

5 Numerical results

Our local linearity strategy was implemented with the classical fourth-order Runge-Kutta method as a basic method. In all the simulation reported in this subsection we used the following adaptivity parameters: $\tau = 10^{-3}$, $s_1 = 0.6$, and $s_2 = 3$. First of all, in figure 4 we show the

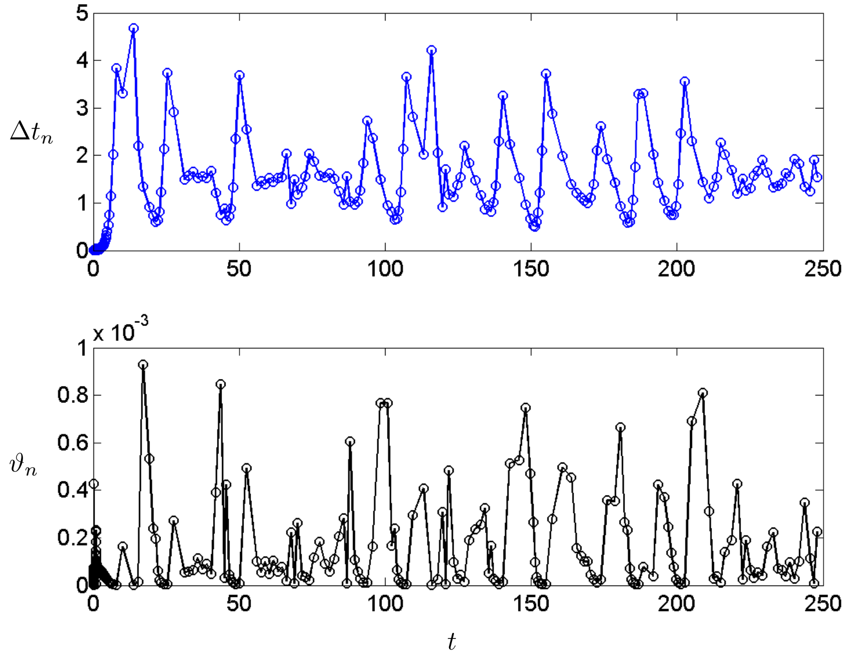

Figure 5: Local linearity control. Top frame: step selection. Bottom frame: monitor function ϑ_n.

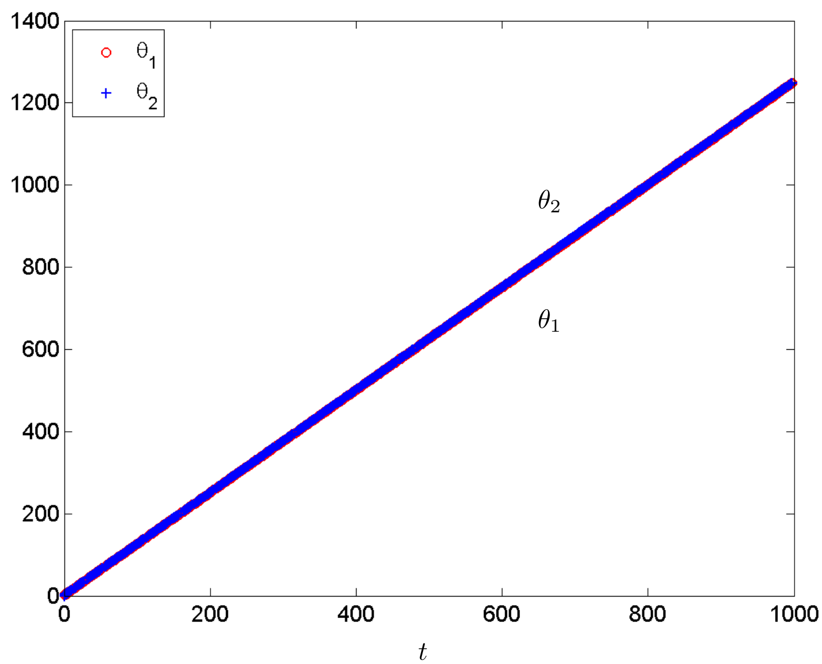

Figure 6: Local linearity control. Longer computation.

numerical results within the range $t \in [0, 250]$. Our algorithm used 452 successful steps plus 27 rejections to compute the numerical solution. The used step sizes were included within the limits $\Delta t_{\min} \approx 1.21 \cdot 10^{-8}$ and $\Delta t_{\max} \approx 4.67$. Note that our $\Delta t_{\max}$ is more than three times larger the stability limit 1.44 suggested by Skufca

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[16]. Figure 5 shows the step size selection and the monitor function defined by equation (5), with the initial step taken by enforcing (6). As far as the value of ϵ is concerned, we used the MATLAB rounding off unit `eps`, that is $\epsilon \approx 2.2205 \cdot 10^{-16}$. We had to define a tentative initial step, so that we used $\Delta t_0 = 10$. However, the proposed value of the initial step size, as it is easily seen from figure (5), has been reduced by our adaptive algorithm in order to complain with the user defined tollerance τ.

On the previous page, figure 6 shows a longer integration where $t \in [0, 1000]$. Let us remark here that, even for this range of the time variable, our numerical trajectories of the two oscillators do not decorrelate. Our algorithm used 894 successful steps plus 39 rejections to compute the numerical solution shown in figure 6.

6 Conclusions

It seems that, as suggested by Skufca, the crux of the `ODE45` method is the step-size update formula. However, our point of view is that neither the analysis of section 3 nor the one in section 4 in Skufca paper are really valid. In fact, his argument is that being each attracting solution component of the system a straight line, a Runge-Kutta method of any order should be exact in approximating the slope and the error estimate has to tend to zero, so that the selected step size will increase until it becomes large enough to make the method unstable. The above reasoning has a weak point: it is only valid for step-sizes tending to zero but it makes a conclusion for large step sizes. Furthermore, it does not take into account the rounding errors of any real computation. Moreover, the same analysis can be applied to the control based on local linearity strategy, but, as we have reported in the previous section, with this adaptive strategy we have obtained a correlated solution within the domain $t \in [0, 1000]$.

References

[1] C. J. Budd, B. Leimkuhler, and M. D. Piggott. Scaling invariance and adaptivity. *Appl. Numer. Math.*, 39:261–288, 2001.

[2] C. J. Budd and M. D. Piggott. The geometric integration of scale invariant ordinary and partial differential equation. *J. Comput. Appl. Math.*, 128:399–422, 2001.

[3] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations, Runge-Kutta and General Linear Methods.* Whiley, Chichester, 1987.

[4] J. R. Dormand and P. J. Price. A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, 6:19–26, 1980.

[5] W. H. Enright. A new error-control for initial value solvers. *Appl. Math. Comput.*, 31:588–599, 1989.

[6] W. H. Enright. Continuous numerical methods for ODEs with defect control. *J. Comput. Appl. Math.*, 125:159–170, 2000.

[7] R. Fazio. Adaptive strategies for numerical IVPs solvers. In S.I. Ao, L. Gelman, D.W.L. Hukins, A. Hunter, and A.M. Korsunsky, editors, *Proceedings World Congress on Engineering 2009, held in London, 1-3 July*, volume II, pages 1213–1218. IAENG, Kwun Tong – Hong Kong, 2009.

[8] E. Fehlberg. Classical fifth-, sixth-, seventh- and eighth order formulas with step size control. *Computing*, 4:93–106, 1969.

[9] A. Iserles. *A First Course in the Numerical Analysis of Differential Equations.* Cambridge University Press, Cambridge, 1996.

[10] A. Jannelli and R. Fazio. Adaptive stiff solvers at low accuracy and complexity. *J. Comput. Appl. Math.*, 191:246–258, 2006.

[11] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems.* Wiley, Chichester, 1991.

[12] D. Sarafyan. Error estimation for Runge-Kutta methods through pseudoiterative formulas. Techn. Rep. No 14, Lousiana State Univ., New Orleans, 1966.

[13] L. F. Shampine. Error estimates and control for ODEs. *J. Sci. Comput.*, 25:3–16, 2005.

[14] L. F. Shampine and A. Witt. A simple step size selection algorithm for ODE codes. *J. Comput. Appl. Math.*, 58:345–354, 1995.

[15] L.F. Shampine. *Numerical Solution of Ordinary Differential Equations.* Chapman & Hall, New York, 1994.

[16] J. D. Skufca. Analysis still matters: a surprising instance of failure of Runge-Kutta-Felberg ODE solvers. *SIAM Review*, 46:729–737, 2004.

[17] S. Strogatz. *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering.* Perseus Books, Cambridge, MA, 1994.

[18] J. M. Verner. Explicit Runge-Kutta methods with estimates of the local truncation error. *SIAM J. Numer. Anal.*, 15:772–790, 1978.

ISBN: 978-988-18210-8-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)