Adaptive Strategies for Numerical IVPs Solvers

Riccardo Fazio [*]

Abstract—**This paper is concerned with adaptive numerical methods for initial value problems governed by systems of ordinary differential equations. Here we introduce a novel step selection algorithm based on the simple idea that locally all continuous functions can be suitably approximated by a straight line. Finally we present two sample numerical computations performed by our step selection algorithm.**

Keywords: initial value problems, adaptive numerical methods, local error control approaches, stability approach, 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

$$\begin{cases} \dfrac{du}{dt} = \mathrm{f}(u) \ , \qquad t \in [t_0, t_{\max}] \\ u(t_0) = u_0 \ , \end{cases} \tag{1}$$

where $u(t) : \mathbb{R} \to \mathbb{R}^k$, $u_0 \in \mathbb{R}^k$ and $\mathrm{f}(u) : \mathbb{R}^k \to \mathbb{R}^k$.

We note that any adaptive algorithm should put more grid points in the regions where the solution undertakes a great variability. Accepted strategies for variable step size selection are based mainly on the inexpensive monitoring of the local truncation error, or 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 [16, pp. 107-109] or [14, pp. 75-81]. Extensions to embedded Runge-Kutta methods have been developed by Sarafyan [17], Fehlberg [12], Verner [21] and Dormand and Price [8];

2. local error control based on Richardson local extrapolation, see Bulirsch and Stoer [6] and the treatment reported by Hairer et al. [13, pp. 228-233];

3. residual (or size of the defect) monitoring, proposed by Enright [9], see also his survey paper [10];

4. monitoring the relative change in the numerical solution, proposed by Shampine and Witt [19] and recently justified by Jannelli and Fazio [15]. It will be shown below that this approach is related to monitoring the stability of the numerical computation.

5. adaptivity by scaling invariance, proposed for the numerical solution of blow-up problems by Budd et al. [4, 5].

The main aim of this work is to present a new strategy based on monitoring the approximate local linearity of the computed solution. This is a novel approach, which is based on the simple idea that locally every continuous function behaves like a linear function.

In the next section we describe some of the approaches listed above, explaining in full details our linearity monitoring.

2 Adaptive step size strategies

Given a step size Δt_n and an initial value u_n at time t_n, a one-step method computes an approximation u_{n+1} at time $t_{n+1} = t_n + \Delta t_n$. Henceforth, any adaptive strategy deals with a prescribed criterion in order to define or modify the current step size Δt_n.

2.1 Local error control and Adaptivity

The most common used strategy for the adaptive numerical solution of IVPs is to take into account some kind of local truncation error control. In the following we recall, briefly, how this is done in the case of embedded Runge-Kutta methods. The local error (LE) is defined as the error made by the numerical method within a single step

$$LE_{n+1} = w(t_{n+1}) - u_{n+1}$$

[*] 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: March 19, 2009.

where $w(t_{n+1})$ is the exact solution of the reference IVP

$$\left\{ \begin{array}{l} \dfrac{dw}{dt} = \mathrm{f}(w) \\ w(t_n) = u_n \ . \end{array} \right. \tag{2}$$

Note that for a method of order p, we have $LE = O(\Delta t^{p+1})$. Many routines make use of two methods of different order to estimate the LE: a method of order p computes a first approximation u_{n+1} and a method of order $\tilde{p} > p$ provides a more accurate approximation $\tilde{u}_{n+1}$, so that, we can define

$$\begin{array}{rcl} est & = & \tilde{u}_{n+1} - u_{n+1} \\ & = & \tilde{u}_{n+1} - w(t_{n+1}) + [w(t_{n+1}) - u_{n+1}] \\ & = & O\left(\Delta t^{p+2}\right) + LE_{n+1} \ , \end{array}$$

here we infer that $\tilde{LE}_{n+1} = \tilde{u}_{n+1} - w(t_{n+1})$ is at least $O\left(\Delta t^{p+2}\right)$ because we assumed that $\tilde{p} > p$. Henceforth, *est* is an asymptotically correct estimate of the LE for the lower order method.

The relationship between local and global error (GE) is given by

$$\begin{array}{rcl} GE_{n+1} & = & u(t_{n+1}) - u_{n+1} \\ & = & u(t_{n+1}) - w(t_{n+1}) + LE_{n+1} \ . \end{array}$$

$u(t_{n+1}) - w(t_{n+1})$ is the exact solution of the governing system with initial data given by

$$\begin{array}{rcl} u(t_n) - w(t_n) & = & u(t_n) - u_n \\ & = & GE_n \ , \end{array}$$

so that, it depends on the analytical stability of the governing system. In other words, at each step the GE is made by two addends: the GE evolved by the governing system plus the current LE. If the IVP is unstable, then, no matter the order of the LE, any small error, like roundoff errors, will grow within the numerical solution. On the contrary, in the case of a stable IVP, the GE evolution would be damped, that is

$$\| u(t_{n+1}) - w(t_{n+1}) \| \leq \| GE_n \| \ ,$$

and the global error of a method of order p will be an $O(\Delta t^p)$.

2.2 Extrapolation and Adaptivity

A different way to provide an estimate of the LE is to apply the same method with two different time steps. The simplest case is to use Δt_n and $\Delta t_n/2$. We can compute the solution $y_{n+1}(\Delta t_n)$ by a single step and $y_{n+1}(\Delta t_n/2)$ by two half-steps. With a method of order p we can define the extrapolated value

$$\hat{y}_{n+1} = \frac{\left(\frac{1}{2}\right)^p y_{n+1}(\Delta t_n) - y_{n+1}\left(\Delta t_n/2\right)}{\left(\frac{1}{2}\right)^p - 1}$$

which has an order of accuracy $p+1$. Now, an extrapolated estimation of the LE can be obtained as

$$\begin{array}{rcl} ext & = & \hat{y}_{n+1} - y_{n+1} \\ & = & \hat{y}_{n+1} - u(t_{n+1}) + u(t_{n+1}) - y_{n+1} \\ & = & O\left(\Delta t^{p+2}\right) + LE_{n+1} \ . \end{array}$$

In this case, *ext* is an asymptotically correct estimate of the LE of the method used with the largest step size.

2.3 Stability and Adaptivity

In this sub-section, we recall the simple adaptive procedure defined by Jannelli and Fazio [15]. This procedure applies the monitor function

$$\eta_n = \frac{\| y_{n+1} - y_n \|}{\Gamma_n} \ , \tag{3}$$

where Γ_n is defined by

$$\Gamma_n = \left\{ \begin{array}{ll} \| y_n \| & \text{if} \quad \| y_n \| \neq 0 \\ \epsilon & \text{otherwise} \ , \end{array} \right.$$

with $0 < \epsilon \ll 1$, and requires that the step size will be modified as needed in order to keep η_n between chosen tolerance bounds, say $0 < \eta_{\min} \leq \eta_n \leq \eta_{\max}$. We would like to relate, now, this kind of monitor function to the concept of stability and to this end we start by recalling the definition of the asymptotic stable numerical method. For a asymptotic stable method we require that

$$\| u_{n+1} \| \leq \| u_n \| \ .$$

Now, it is a simple matter to show that

$$\| u_{n+1} \| - \| u_n \| \leq \| u_{n+1} - u_n \|$$

so that, with the above adaptivity criterion we also require that

$$\| u_{n+1} \| - \| u_n \| \leq \eta_{\max} \Gamma_n \ .$$

We remark here that the asymptotic stability requirement can be enforced by letting $\eta_{\max}$ go to zero.

2.4 Linearity and Adaptivity

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

$$\vartheta_n = \frac{r}{1+r} \max_{j=1,\ldots,k} \left| \frac{{}^j u_{n+1} - (1+r) \, {}^j u_n + r \, {}^j u_{n-1}}{{}^j u_n + \epsilon} \right| \tag{4}$$

where $r = \Delta t_n / \Delta t_{n-1}$, ϵ is defined as before, and the notation for the components of a vector introduced by Lambert [16, p. 3] has been used, so that ${}^j u$ is the j-th component of the vector u. Again, we can require that the step size selection is such that ϑ_n satisfies the conditions $0 < \vartheta_{\min} \leq \vartheta_n \leq \vartheta_{\max}$, where $\vartheta_{\min}$ and $\vartheta_{\max}$ are tolerance bounds.

3 Step size selection

For the current step size selection the user should specify a suitable norm and a tolerance τ. It turns out that the two cases of a local error control made by two different order methods or an extrapolation algorithm can be treated together. In fact, our aim would be to get

$$\|err\| \leq \tau$$

where err can be est or ext. It has been shown by Shampine [18] that, both in the case when the step is rejected and repeated or in the case of a successful step, the largest factor α to multiply by the current step size in order to get the next a successful one is given by

$$\alpha = \left(\frac{\tau}{\|err_n\|} \right)^{1/(p+1)} .$$

Henceforth, we can use the new step size $\Delta t_{n+1} = \alpha \Delta t_n$.

As far as the stability and the linearity strategies described above are concerned, their step size selection can be treated together. The basic guidelines for setting the step size are given by the algorithm defined by the following steps:

1. Given a step size Δt_n and an initial value y_n at time t_n, the method computes a value y_{n+1} and, consequently, a monitoring function η_n by the above formula;

2. If $\eta_{\min} \leq \eta_n \leq \eta_{\max}$, then t_n is replaced by $t_n + \Delta t_n$; the step size Δt_n is not changed and the next step, subject to the check at Step 6, is taken by repeating Step 1 with initial value y_n replaced by y_{n+1};

3. If $\eta_n < \eta_{\min}$, then t_n is replaced by $t_n + \Delta t_n$ and Δt_n is replaced by $\rho \Delta t_n$ where $\rho > 1$ is a step size amplification factor; the next integration step, subject to the checks at Step 5 and Step 6, is taken by repeating Step 1 with initial value y_n replaced by y_{n+1};

4. If $\eta_n > \eta_{\max}$, then t_n remains unchanged; Δt_n is replaced by $\sigma \Delta t_n$, where $0 < \sigma < 1$ and the next integration step, subject to the check at Step 5, is taken by repeating Step 1 with the same initial value y_n;

5. If $\Delta t_{\min} \leq \Delta t_n \leq \Delta t_{\max}$, return to Step 1; otherwise Δt_n is replaced by $\Delta t_{\max}$ if $\Delta t_n > \Delta t_{\max}$ or by $\Delta t_{\min}$ if $\Delta t_n < \Delta t_{\min}$, then proceed with Step 1.

6. If $t_n > t_{\max}$, then we set $t_n = t_{\max}$, and $\Delta t_n = t_{\max} - t_{n-1}$.

So that the user have to define the following values:
Δt_0 the initial step size;
$\Delta t_{\min}$ minimum value of the step that can be used;
$\Delta t_{\max}$ maximum value of the step that can be used;
ρ step amplification factor;
σ step reduction factor;
$\eta_{\min}$ lower bound for the tolerance;
$\eta_{\max}$ upper bound for the tolerance.

Note that, in the case of linearity monitoring, we have to replace η_n, $\eta_{\min}$, and $\eta_{\max}$ by ϑ_n, $\vartheta_{\min}$, and $\vartheta_{\max}$, respectively. Moreover, the stability and linearity adaptive approaches are based on implementing only one numerical method, and, in order to advance the computation, they use two and three numerical approximations obtained at two and three consecutive time steps, respectively.

4 Test problems and numerical results

In this section we consider a test problem and report related numerical results.

The classical two-body problem is a Kepler problem in which, for instance, a small body like a satellite or a comet orbits around the sun. Since the gravitational force is a central one the motion belongs to the plane defined by the two bodies and the satellite initial velocity. We can use a Copernican coordinate system and fix the sun at its origin. The problem is governed by the following IVP

$$\begin{cases} \dfrac{d^2\mathbf{r}}{dt^2} = -\dfrac{GmM}{|r|^3}\mathbf{r} \\ \mathbf{r}(0) = \mathbf{r}_0 \ , \quad \text{and} \quad \dfrac{d\mathbf{r}}{dt} = \mathbf{v}(0) = \mathbf{v}_0 \ , \end{cases} \tag{5}$$

where G is the universal gravitational constant, etc.

The total energy E of the satellite is given by

$$E = \frac{1}{2}mv^2 - \frac{GMm}{r} \ , \tag{6}$$

where $r = |\mathbf{r}|$, and $v = |\mathbf{v}|$. This total energy is conserved as is the angular momentum

$$\mathbf{M} = \mathbf{r} \times m\mathbf{v}$$

their values at all time are equal to the values computed at the initial time.

We can rewrite the model (5) in the general form (1) by setting

$$\mathbf{u} = (x, y, v_1, v_2)^T$$

$$\mathbf{f} = \left(v_1, v_2, -\frac{GmM}{\sqrt{x^2+y^2}(x^2+y^2)}x, -\frac{GmM}{\sqrt{x^2+y^2}(x^2+y^2)}y \right)^T$$

$$\mathbf{u}_0 = (x(0), y(0), v_1(0), v_2(0))^T$$

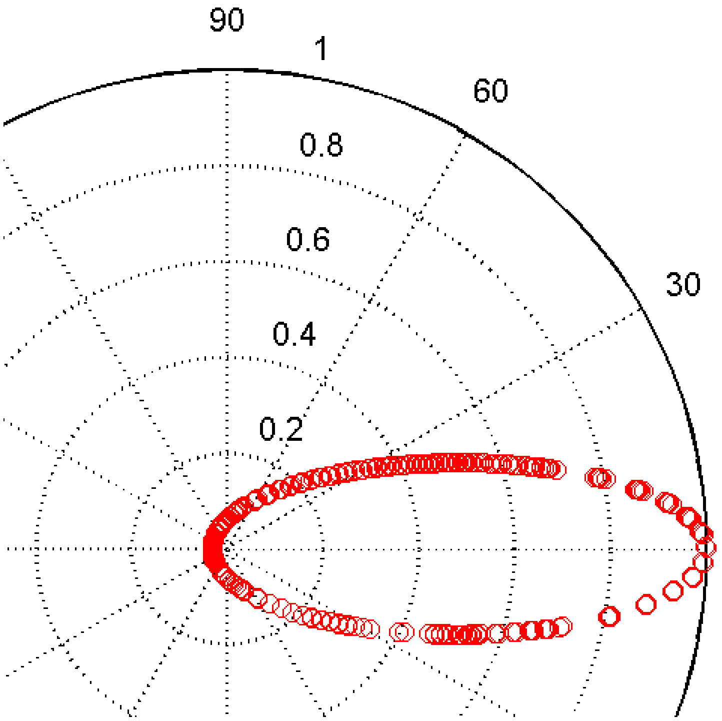

Figure 1: Linearity control. Seven orbits computation.

For this kind of problems the natural units for length and time are the astronomical units (AU): 1 AU $= 1.496\times10^{11}$ m, which equals the mean Earth-Sun distance; and the AU year: the period needed to travel a circular orbit of radius 1 AU, respectively.

In the following, as a specific test problem, we consider the following parameters: the mass of the satellite is taken as unitary, the mass of the larger body is $M = 1.99 \times 10^{30}$ kg, and $G = 6.67 \times 10^{-11}$ $\mathrm{m}^3/(\mathrm{kg}\cdot \mathrm{s}^2)$ is the universal constant. Moreover, we use the following initial conditions: the initial position is given by $x(0) = 1$ AU, $y(0) = 0$, and the initial velocity is $v_1(0) = 0$, $v_2(0) = \pi/2$ AU/yr .

For the simulations reported below we used, as a basic one-step method, the classical fourth order Runge-Kutta method, see for instance Butcher [7, p. 166].

4.1 Linearity control

First of all, in figure 1 we show the numerical results of a successful computation. These results were obtained by setting the following adaptivity parameters: $\eta_{\min} = 0.01$, $\eta_{\max} = 10 \cdot \eta_{\min}$, $\rho = 4$, and $\sigma = 0.25$. Moreover, by setting $\Delta t_0 = 0.02$ our algorithm used 749 successful steps plus 218 rejections to compute seven complete orbits. The used step sizes were included within the range $\Delta t_{\min} = 7.8125{\cdot}10^{-5}$ and $\Delta t_{\max} = 0.02$. The total energy given by equation (6) was conserved along the computation, see figure 2. Figure 3 shows the step size selection and the monitor function defined by equation (4). From figure 3 it might seem that we have chosen an *ad hoc* value for the initial time step Δt_0. However, it is worth noticing that a large selection of Δt_0 will be reduced by the adaptive algorithm suggesting a suitable value of the initial time step. This remark can be understood by looking at figure 4.

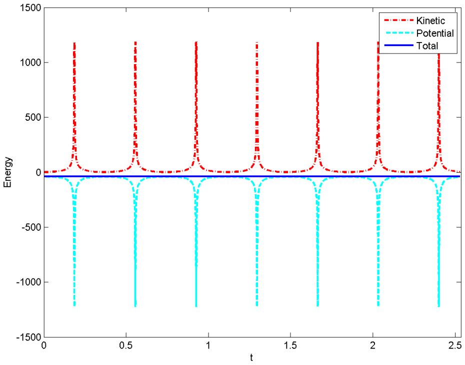

Figure 2: Linearity control. Total, potential and kinetic energies plot with a total energy conservation.

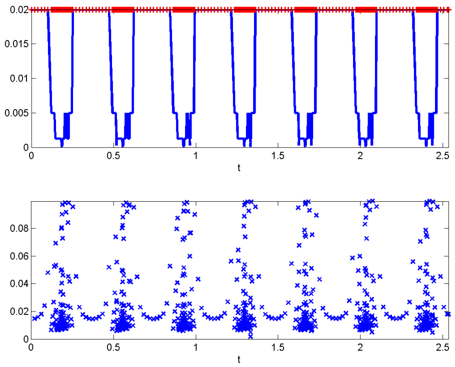

Figure 3: Linearity control. Top frame: step selection. Bottom frame: monitor function.

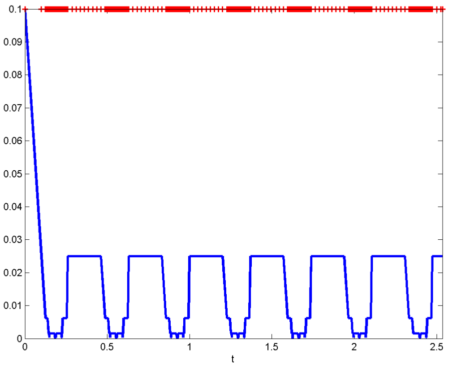

Figure 4: Linearity control. Step size selection for a large value of Δt_0.

Next, we report on an inaccurate selection of the adaptivity parameters. Figure 5 shows a typical case. For this computation we used $\eta_{\min} = 0.05$, again $\eta_{\max} = 10 \cdot \eta_{\min}$, and $\Delta t_0 = 0.001$. Let us remark here that the inaccuracy of the above numerical results can be realized by looking at figure 6 where the total energy along with the potential and kinetic energies are plotted. As it is easily seen the total energy is not conserved in this computation.

5 Conclusions

In the introduction we listed the adaptive step size strategies that have been used in literature. Then, we introduced a novel step selection algorithm based on the simple idea that locally all continuous functions can be suitably approximated by a straight line. We remark that the stability and the linearity approaches can be used anytime the complexity of the problem to be solved calls for a low complexity numerical implementation of the adaptive procedure, see the motivations reported by Jannelli and Fazio [15] or by Vergara and Zunino [20]. On the other hand, there are differences between these two strategies. First of all, the stability monitor function (3) is a one-step criterion whereas the linearity monitor function (4) is a two-steps one. Furthermore, the formula for η_n is not invariant with respect to scaling transformations, whereas on the contrary ϑ_n is invariant under rescaling. This scaling invariant property is of interest anytime we use scaling invariance for solving a challenging problem by rescaling the solution of a simpler, from a numerical viewpoint, problem. As an explicative example, see the recent treatment of the van der Pol model by Fazio [11].

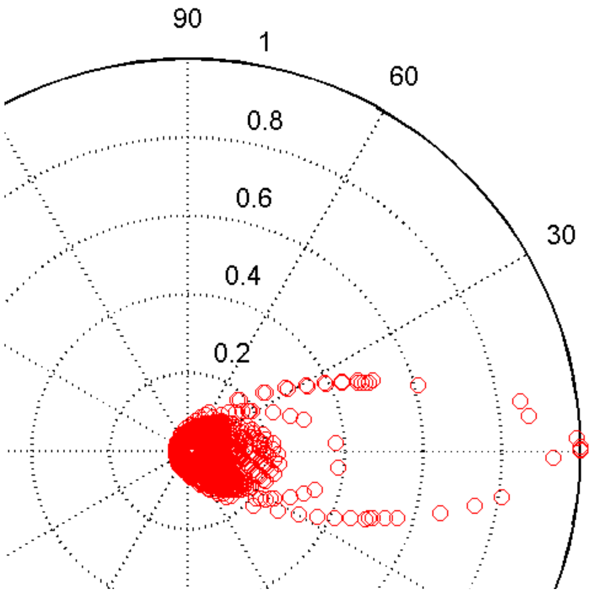

Figure 5: Linearity control. Not periodic orbit computation.

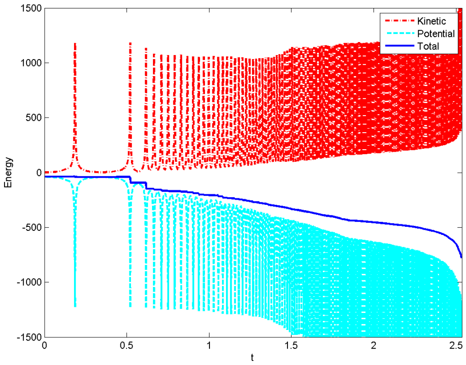

Figure 6: Linearity control. Not conservative energy plot.

Our main topic, that is the adaptive implementation of numerical methods, is a fundamental one also for the numerical solution of partial differential problems. As an example, the adaptive approach based on scale invariance properties, cited in the introduction, has been extended to problems with blow-up solutions governed by parabolic models by Budd and his co-workers [3, 2, 1].

References

[1] C. J. Budd and G. J. Collins. An invariant moving mesh scheme for the nonlinear diffusion equation. *Appl. Numer. Math.*, 26:23–39, 1998.

[2] C. J. Budd and G. J. Collins. Symmetry based numerical methods for partial differential equations. In D. F. Griffiths, D. J. Higham, and G. A. Watson, editors, *Numerical Analysis 1997*, pages 16–36, Harlow, 1998. Longman. Pitman Research Notes in Mathematics 380.

[3] C. J. Budd, W. Huang, and R. D. Russell. Moving mesh methods for problems with blow-up. *SIAM J. Sci. Comp.*, 17:305–327, 1996.

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

[5] 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.

[6] R. Bulirsch and J. Stoer. Numerical treatment of ordinary differential equations by extrapolation methods. *J. Comput. Appl. Math.*, 6:19–26, 1966.

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

[8] J. R. Dormand and P. J. Price. A family of embedded Runge-Kutta formulae. *Num. Math.*, 8:1–13, 1980.

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

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

[11] R. Fazio. Numerical scaling invariance applied to the van der Pol model. *Acta Appl. Math.*, 104:107–114, 2008.

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

[13] E. Hairer, S. P. Norsett, and G. Wanner. *Solving Differential Equations I: Nonstiff Problems.* Springer, Berlin, 1987.

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

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

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

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

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

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

[20] C. Vergara and P. Zunino. Multiscale boundary conditions for drug release from cardiavascular stents. *Multiscale Model. Simul.*, 7:565–588, 2008.

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