

Time-step selection algorithms: Adaptivity, Control, and Signal Processing

Gustaf Söderlind ¹

*Numerical Analysis, Centre for Mathematical Sciences, Lund University, Box
118, S-221 00 Lund, Sweden (Gustaf.Soderlind@na.lu.se)*

Abstract

The efficiency of numerical time-stepping methods for dynamical systems is greatly enhanced by automatic time step variation. In this paper we present and discuss three different approaches to step size selection: (i) *control theory* (to keep the error in check); (ii) *signal processing* (to produce smooth step size sequences and improve computational stability); and (iii) *adaptivity*, in the sense that the time step should be covariant or contravariant with some prescribed function of the dynamical system's solution. Examples are used to demonstrate the different advantages in different applications. The main ideas are further developed to approach some open problems that are subject to special requirements.

Key words: Step size selection, control, adaptivity, signal processing, reversible systems

1 Introduction

Numerical time-stepping methods for dynamical systems are typically implemented with various support algorithms in order to increase efficiency. “Adaptivity” is the most important: one needs to strike a balance between computational error and effort, usually based on local considerations, as solution properties may vary most significantly along the computed trajectory. By varying the step size, efficiency may sometimes be increased by several orders of magnitude without a significant loss of accuracy.

We shall assume that the problem to be solved is a dynamical system

$$\dot{y} = F(y); \quad y(0) = y_0. \quad (1)$$

¹ Supported by the Swedish Research Council under contract VR 2002-5370.

where $y \in \mathbb{R}^d$ and $F : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a smooth Lipschitz map. Let us for the time being assume that we apply an explicit or implicit one-step method to (1). Thus, given a step size h , there is a map $\Phi_h : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that

$$y_{n+1} = \Phi_h(y_n); \quad y(0) = y_0, \quad (2)$$

is a discrete-time dynamical system approximating the flow of (1) on \mathbb{R}^d . To vary the step size, we need an additional map $\Psi_y : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$h_{n+1} = \Psi_y(h_n). \quad (3)$$

The subscript y indicates that the step size recursion uses information about the numerical solution $\{y_n\}$ in the construction of h_{n+1} . An adaptive time-stepping method is the interactive recursion

$$y_{n+1} = \Phi_{h_n}(y_n) \quad (4a)$$

$$h_{n+1} = \Psi_{y_{n+1}}(h_n). \quad (4b)$$

It is immediately clear that stability is affected: (4) is an augmented dynamical system on \mathbb{R}^{d+1} , while the constant step size method (2), which has $\Psi_y \equiv I$, is only a dynamical system on \mathbb{R}^d . Other aspects of the dynamics may however also be affected.

In practical computations, the time-stepping method (2) is usually equipped with an embedded error estimate. In the asymptotic regime ($h \rightarrow 0$) the *step size – error relation* is assumed to be

$$r_n = \hat{\varphi}_n h_n^k, \quad (5)$$

where r_n is the norm of the local error estimate, $\hat{\varphi}_n$ is the norm of the principal error function, and k is related to the method order. The model of the step size – error relation is crucial to the design of efficient controllers for adaptive step size selection, [4,5,17,18]. The simplest and most common candidate for a local error control law is to choose

$$h_{n+1} = \left(\frac{\varepsilon}{r_n} \right)^{1/k} h_n, \quad (6)$$

where ε is (a fraction of) the local error tolerance, [7,17,18,2, p. 156]. This elementary control law has been suggested as it “immediately” eliminates a deviation between r_n and ε if the method operates in the asymptotic regime and the principal error function $\hat{\varphi}_n$ varies slowly.

Although other choices may be preferred in special applications, cf. [8,9,21], in the elementary controller (6), Ψ_y is a linear map. However, in practice the step selection algorithm based on (6) is often implemented as a discontinuous, nonsymmetric and nonlinear map, [19]. Extra conditions and logic are

added to offer what is assumed, but not proved, to be increased “safety” and “efficiency.” A rigorous analysis of (4) then becomes impossible. In addition, one also observes a loss of computational stability in practical computations, contrary to classical priorities and requirements in numerical computing, [19].

By instead choosing step size selection algorithms that are supported by analysis, one can distinguish at least three different approaches: *control theory*, *signal processing*, and *adaptivity* based on the assumption that the step size should follow some prescribed function of the solution.

Control theory. Using a control theoretic approach, the objective is to *keep the estimated error equal to the tolerance*. In order to do this, one needs to measure the difference between the error and target tolerance, and use this deviation in a *feedback control system* to continually correct the step size.

In principle, this is what (6) does, [17]. However, less fortunate implementation practices as well as problems observed in computations led to a search for alternative controllers, with early attempts found in [3,22,23]. Hall and Higham [10–13] studied the oscillatory instabilities when explicit one-step methods encountered mildly stiff ODE problems. A full control theoretic analysis, including the design of new proportional–integral (PI) controllers for explicit as well as implicit Runge–Kutta methods, first appeared in [4,5]. Related problems, e.g. the coordination with iterative nonlinear solvers [6], the construction of suitable predictors [14] and iteration termination criteria [15,19], as well as the problem of finding appropriate error estimators [16] were also studied. The use of control theory in the ODE area was reviewed in [17].

Signal processing. Closely related to control theory, the signal processing approach shifts the emphasis towards producing a smooth step size sequence. The aim is to *create a regular step size sequence from the error estimates, while keeping the error sufficiently close to the tolerance*.

This approach was introduced in [18], which also introduces the necessary systems theory for the design of causal digital filters for step size generation. The error estimates form a discrete-time input signal, from which a regular output signal, the step size sequence, is derived. Regularity means that the sequence has the appearance of samples drawn from a smooth signal, and can be analyzed in terms of spectral content. Tests with well-known codes show that a regular step size sequence has a significant impact on the computational stability of a code, see [19,20], without incurring extra computational costs.

Adaptivity. Although both approaches described above are commonly termed adaptive, the step size sequence can also be “adapted” to a prescribed function of the solution $\{y_n\}$ without direct control of errors or regularity. For example, in geometric integrators the primary concern is to *preserve structure* such as first integrals and reversibility, [8].

In [21] an implicit step size selection algorithm is introduced, while [9] offers an explicit, stable, reversible step size selection algorithm that offers enhanced accuracy and efficiency for reversible systems. Based on *step density control*, this approach automatically produces smooth step size sequences, while structural properties are preserved.

This adaptive approach augments (1) by an additional differential equation for the determination of the step density. In some cases, the very same integration method that is used to solve the problem can be used as the discrete controller that selects the step size by solving the step density differential equation. Such methods become *auto-adaptive*.

Structure of the paper. This paper reviews some of the recently developed techniques in adaptive time-stepping. The main ideas are presented without requiring extensive knowledge of control theory. New examples illustrate the techniques (such as digital filters) as well as applications in geometric integration. We also develop some of these ideas in a few examples on how to approach multistep methods: multistep density control for geometric integration is demonstrated, and closed loop (controller dependent) 0-stability of the BDF3 method is investigated in numerical experiments. The paper is intentionally open-ended and concludes by commenting on open problems in different areas of differential equations, and what ideas might be used to address those problems in future research.

2 Control and signal processing

In the context of step size control, signal processing and control theory are closely related. The control objective is to design an algorithm that keeps

$$r_n \approx \theta \cdot TOL. \quad (7)$$

This objective is achieved by varying h_n . As the asymptotic model suggests that $r_n = \varphi_n h_n^k$, the step size h obviously responds to variations in the principal error function φ . Adaptivity is therefore concerned with the (logarithmic) step size map

$$H_\varphi : \log \varphi \mapsto \log h. \quad (8)$$

Here $\log \varphi$ denotes the entire sequence $\{\log \varphi_n\}$, while $\log h$ denotes $\{\log h_n\}$. The “data” $\log \varphi$ may be considered to consist of a deterministic “signal,” possibly distorted by some stochastic “noise.” The signal processing aspect is to provide sufficient noise abatement to extract the signal, with a restrained violation of the control objective (7). This can be achieved with digital filters.

Signal processing and digital filters. In our context a (causal) linear digital

filter is a *linear difference equation*

$$M(q) \log h_n = N(q) \log \varphi_n, \quad (9)$$

where M and N are polynomials in the forward shift operator q . We consider the forcing function of the right-hand side, $\log \varphi$, to be the data that produce the solution $\log h$. Stability is a requisite: homogeneous solutions are supposed to damp out in a few steps but are of little further interest.

The step size map (8) is concerned with *particular solutions* when the data $\log \varphi$ is a bounded sequence. The difference equation (9) can then be analyzed in terms of its z *transform* (discrete-time Laplace transform), i.e., the rational function $N(q)/M(q)$.

In particular, we are interested in spectral properties and consider (quasi) periodic input by taking $\log \varphi_n = e^{i\omega n}$ for $\omega \in [0, \pi]$. Because of linearity, with this input (9) produces an output $\log h_n = A(\omega)e^{i\omega n}$, where

$$A(\omega) = \frac{N(e^{i\omega})}{M(e^{i\omega})}. \quad (10)$$

Hence, in the frequency domain, the step size map is described by

$$H_\varphi : \{e^{i\omega n}\} \mapsto A(\omega)\{e^{i\omega n}\}, \quad (11)$$

where the amplitude $|A(\omega)|$ measures the attenuation of the oscillation $\{e^{i\omega n}\}$. Recall that $\omega \in [0, \pi]$ and note that $A(\pi) = 0$ if $N(-1) = 0$. Choosing $N(q)$ such that it has a zero at $q = -1$ therefore *blocks signal transmission* for $(-1)^n$ oscillations (as $e^{i\pi n} = (-1)^n$). The output signal $\log h$ is then *smoother* than $\log \varphi$ as it no longer contains the jagged $(-1)^n$ oscillation, which has been filtered out. Other frequencies ω can be filtered out in a similar manner by putting zeros of $N(q)$ at $e^{i\omega}$ on the unit circle.

To summarize, the behavior of a digital filter is governed by the poles and zeros of the z transform $N(q)/M(q)$. Poles are for stability reasons required to remain well inside the unit circle, see (9). Zeros on the unit circle filter out the corresponding frequency and strongly damp nearby frequencies. Just like in the Riemann–Lebesgue lemma, a signal is smoother the smaller its high frequency content is, and blocking the highest frequency ($\omega = \pi$) by putting a zero at $q = -1$ therefore produces a smoother output.

Control and filtering. The question is now: how do we find a controller that implements the filter? As $\log \varphi$ is not directly available, we have to construct the step size sequence $\log h$ from the observed error estimate sequence $\log r$. The elementary controller (6) provides a simple, intuitive solution:

$$(q - 1) \log h_n = \log(\theta \cdot TOL) - \log r_n.$$

This is, however, neither the most general nor the “best” solution. The general construction is covered by the linear difference equation

$$(q - 1)Q(q) \log h_n = P(q)[\log(\theta \cdot TOL) - \log r_n], \quad (12)$$

where P and Q are polynomials of the same degree (to make the recursion explicit) in the forward shift q . The factor $q-1$ on the left hand side is a forward difference; this is known to be necessary, from the control theoretic point of view, in order to eliminate persistent errors—the step size must eventually be adapted to make that persistent error equal the target value, $\theta \cdot TOL$. For simplicity, we put $\theta \cdot TOL := 1$ below.

The *control law* (12) is now related to the desired digital step size filter (9) via the *asymptotic step size – error model* $r_n = h_n^k \varphi$, i.e.,

$$\log r_n = k \log h_n + \log \varphi_n. \quad (13)$$

When (13) is inserted into (12) we obtain

$$(q - 1)Q(q) \log h_n = -kP(q) \log h_n - P(q) \log \varphi_n. \quad (14)$$

Solving for $\log h$, we find the *closed loop system* corresponding to (9),

$$[(q - 1)Q(q) + kP(q)] \log h_n = -P(q) \log \varphi_n. \quad (15)$$

This identifies the link between the digital filter and the controller: in terms of the controller’s defining polynomials P and Q , the step size map is simply

$$H_\varphi(q) = - \frac{P(q)}{(q - 1)Q(q) + kP(q)}. \quad (16)$$

Thus we see that in the earlier description $H_\varphi(q) = N(q)/M(q)$, the polynomials N and M cannot be considered as independent. The polynomial P which determines the filtering corresponds directly to N , but the denominator M depends on both P and Q ; hence stability will depend on both P and Q .

Filter structures. There are several important filter structures that fit into the control dynamics (12). Some of the interesting cases are:

- *Elementary control.* The common control $h_{n+1} = (\theta \cdot TOL/r_n)^{1/k} h_n$ is obtained with $Q = 1$ and $P = 1/k$.
- *Convolution filters.* Also known as “exponential forgetting,” convolution filters have $Q = 1$ and $P = \gamma < 1/k$. Just like the elementary control, this is a purely integrating controller (I control).
- *PI control.* Proportional–integral control is defined by $\deg P = 1$; $Q(q) = q$.
- *AR filters.* Filters of auto-regressive type require that Q has a zero (or more) at $q = 1$.

- *MA filters.* Filters of moving average type require that P has one or more zeros at $q = -1$.
- *FIR filters.* Finite impulse response filters are characterized by a condition on homogeneous solutions, i.e., $(q - 1)Q(q) + kP(q) \equiv q^\nu$ for some (finite) integer ν . Homogeneous solutions are then identically zero after ν steps. FIR filters are equivalent to deadbeat controllers.

One can also design *noise-shaping filters* that redistribute frequency content to reduce undesired high frequencies that cause non-smooth step size sequences. The filter classes mentioned above are not mutually exclusive but can be combined. For a full account of step size filter design, see [18] which also gives filter coefficient order conditions corresponding to different filter designs.

In our case, filters of MA type are of interest. In the simplest case, when $\deg P = 1$, we choose $P(q) = \beta \cdot (q + 1)$, i.e., a multiple of the averaging operator $(q + 1)/2$. Let us assume that we want a PI controller with this property. We must then choose $Q(q) = q$, and the resulting control is

$$q(q - 1) \log h_n = \beta \cdot (q + 1) [\log(\theta \cdot TOL) - \log r_n]. \quad (17)$$

Taking exponentials yields

$$h_{n+1} = \left(\frac{\theta \cdot TOL}{r_n} \right)^\beta \left(\frac{\theta \cdot TOL}{r_{n-1}} \right)^\beta h_n, \quad (18)$$

where it remains to choose the parameter β . Naturally, β will depend on the method order k , but stability must also be considered. A careful analysis in [18] arrives at the value $\beta = 1/(6k)$, for any order k . Referred to as *H211PI*, this is a PI controller with a first order step size filter, producing smooth step size sequences—Figure 1 shows a simulation run for $k = 4$. The controller is very stable; the roots of the characteristic equation of (15), or equivalently, the poles of the step size map (16), are $q = 1/2$ and $q = 1/3$, respectively. Several other controllers with related properties are given in [18].

3 Step density control and one-step methods

In step density control, a differentiable time transformation $t = \Gamma(\tau)$ with $d\Gamma/d\tau =: 1/\rho(\tau)$ is introduced to locally stretch or compress time. The new independent variable τ is sampled at equidistant points, with $\Delta\tau = \varepsilon$. Then

$$h_{n+1/2} := t_{n+1} - t_n = \Gamma(\tau_{n+1}) - \Gamma(\tau_n) \approx \frac{\varepsilon}{\rho(\tau_{n+1/2})}.$$

Thus, when the *step density* ρ varies, the step size h varies. The staggered grid notation is both convenient and advantageous in the present context.

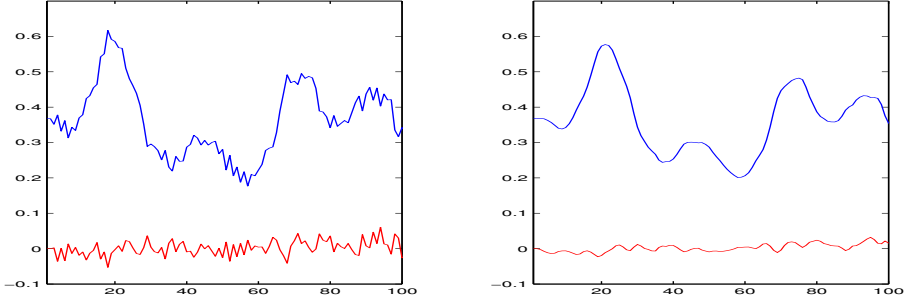


Fig. 1. *Digitally filtered step size sequence.* Simulated step size sequence h is plotted vs step number for 100 steps with the elementary controller (top left) and the $H211PI$ digital filter (top right) responding to identical noisy data φ . The lower graph in both diagrams shows the deviation between the deterministic signal and the actual step size output. The $H211PI$ filter produces a smaller and smoother error; it accurately extracts the deterministic signal and is less sensitive to noise.

As $d/dt = \rho d/d\tau$, the original ODE (1) is transformed into

$$y' = F(y)/\rho, \quad (19)$$

where prime denotes derivative with respect to τ . The time-scaled equation (19) is augmented by the continuous control system

$$\rho' = G(y) \quad (20a)$$

$$t' = 1/\rho, \quad (20b)$$

where (20a) generates ρ via a suitable control function $G(y)$, and (20b) recovers the original time t . The initial values are $\rho(0) = 1$ and $t(0) = 0$, respectively. By solving the augmented system (19–20) numerically, the continuous density $\rho(\tau)$ will be represented by a discrete density sequence $\{\rho_{n+1/2}\}$, where $\rho_{n+1/2} \approx \rho(\tau_{n+1/2})$. Because of the initial value $\rho(0) = 1$, the parameter ε can be interpreted as the initial step size.

Reversible systems. In [9], step density control is applied to reversible systems solved by one-step methods. The target is to keep $\rho = Q(y)$, where Q is a prescribed symmetric function of the solution y . To avoid an implicit step size selection algorithm, this algebraic condition is replaced by the step density generating differential equation (20a).

By taking $G(y) = d(\log Q)/dt = \nabla Q(y)^T F(y)/Q(y)$, the system (20) becomes *Hamiltonian* with first integral $\log[Q(y(t))/\rho]$, i.e., ρ remains proportional to $Q(y(t))$. For numerical computations, however, a *discrete* controller is needed. By using the explicit, symmetric, reversible and symplectic Störmer Verlet method to solve (20a), we obtain the step density controller

$$\rho_{n+1/2} = \rho_{n-1/2} + \varepsilon \nabla Q(y_n)^T F(y_n)/Q(y_n). \quad (21)$$

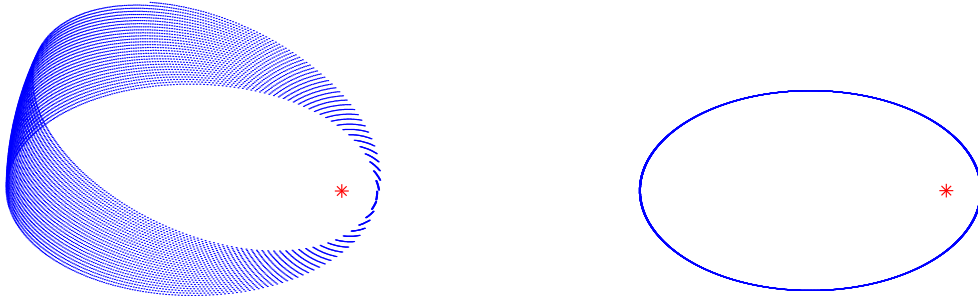


Fig. 2. *Thirty orbits of the Kepler problem computed with the Störmer Verlet method.* With 10,000 constant steps the trajectory shows considerable numerical precession (left). With 10,000 steps of variable size, accuracy is much enhanced and precession strongly suppressed (right). There is near-preservation of the Hamiltonian in both cases, but the adaptive method has approximately 25–30 times smaller energy error.

This generates a density sequence $\{\rho_{n+1/2}\}$ such that $Q(y_n)/\rho_n$ remains nearly constant for exponentially long times (with $\rho_n := (\rho_{n-1/2} + \rho_{n+1/2})/2$), and from which the step size is computed as $h_{n+1/2} = \varepsilon/\rho_{n+1/2}$.

Combining this controller with a symmetric, reversible geometric integrator, applied to an integrable Hamiltonian system, results in a reversible adaptive method that has the desired long term accuracy and stability, see [9]. Figure 2 shows the numerical solution of the Kepler problem, [8, p. 7], for an eccentricity of 0.8. The Störmer Verlet method, [*ibid.*, p. 14], is used, first with constant steps, and then with variable steps taking $G = -p^T q/q^T q$, where q is the position vector and p is the corresponding momentum vector, all in accordance with the technique outlined above. In the latter case, the time step is short near the passage of the central mass and large away from it. Thus the computational effort is spent where the problem is sensitive to perturbations. Consequently, although energy is (nearly) conserved in both cases, accuracy and efficiency are significantly improved by adaptivity.

This reversible adaptivity can be extended to control numerical errors provided that symmetry and reversibility preserving error estimators are employed.

4 Step density control and multistep methods

In general, the use of variable steps with multistep methods is not straightforward. The step density control, however, offers a simple approach: we merely need to solve (19) and (20) using a *constant* step size ($\Delta\tau = \varepsilon$) multistep method. The step density control may, however, affect the *order of convergence* as all variables in (19–20) in general need to be computed to the same order.

Let the method have m steps. The variable grid is generated by solving (20) numerically. In principle, one can construct an auto-adaptive method by letting the same multistep method solve (19) and (20a) simultaneously, but the resulting adaptive method generally becomes implicit. As explicit time step selection schemes are normally preferred, it is natural to consider solving (20a) by some other, suitable method than the one used to solve (19). This step density generator may be an m -step method, but could also have another suitable step number.

The equation (20a) corresponds to a continuous integrating controller. To create a discrete integrating controller, it is replaced by a quadrature formula. One may consider an ε -equidistant, m -step quadrature formula, e.g. of the form

$$\rho_{n+1/2} - \rho_{n-m+1/2} = \varepsilon \sum_{j=0}^{m-1} G(y_{n-j}). \quad (22)$$

Consider $m = 2$ and the controller $\rho_{n+1/2} - \rho_{n-3/2} = \varepsilon[G(y_n) + G(y_{n-1})]$. As the function G is only evaluated at “interior” points of the quadrature interval, the controller is akin to an open Newton–Cotes quadrature formula. Note that a closed formula cannot be used as that would imply that the step size can no longer be generated via an *explicit* recursion. The requirement for explicitness is also the reason why we prefer to describe the control as generating step sizes on a *staggered grid*.

Reversible systems. Controllers of the type (22) may be viewed as multistep generalizations of the Störmer Verlet step density controller (21), obtained for $m = 1$. The quadrature formulas of this type are symmetric; this is necessary for reversible dynamical systems.

In the one-step case we used the control function $G(y) = -p^T q / q^T q$, and we shall now generalize this approach to multistep discretizations. If a multistep method is used to solve the special second order problem $\ddot{q} = f(q)$ by direct discretization of q , there is no approximation to $p = \dot{q}$ available. We may then consider the finite difference approximation

$$-p^T q / q^T q \approx -\frac{2(q_n - q_{n-1})^T (q_n + q_{n-1})}{h_{n-1/2} (q_n + q_{n-1})^T (q_n + q_{n-1})} \quad (23a)$$

$$= -\frac{2\rho_{n-1/2} (q_n - q_{n-1})^T (q_n + q_{n-1})}{\varepsilon (q_n + q_{n-1})^T (q_n + q_{n-1})}. \quad (23b)$$

This suggests that we use the two-step density controller

$$\rho_{n+1/2} - \rho_{n-3/2} = -\frac{4\rho_{n-1/2} (q_n - q_{n-1})^T (q_n + q_{n-1})}{(q_n + q_{n-1})^T (q_n + q_{n-1})}. \quad (24)$$

Note that this controller still satisfies all the necessary symmetry and reversibility conditions around the discretization midpoint, $\tau_{n-1/2}$.

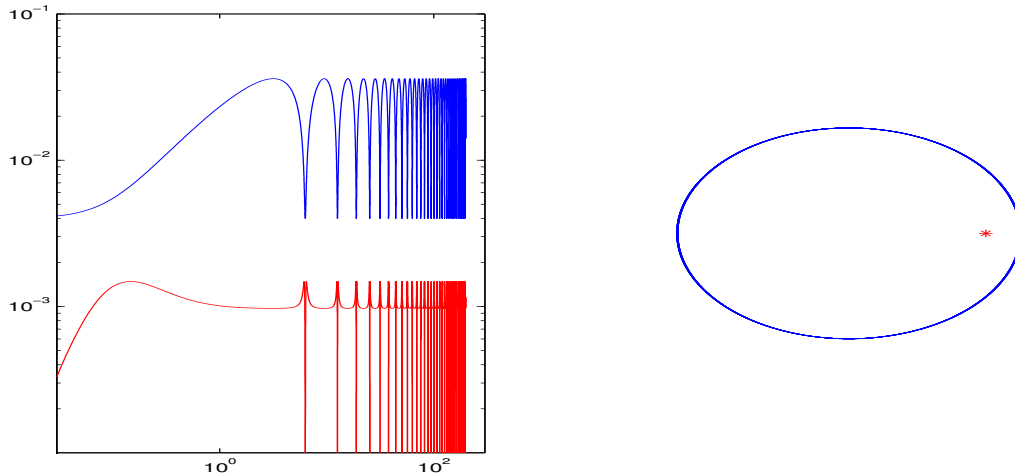


Fig. 3. *Thirty orbits of the Kepler problem computed with the adaptive explicit midpoint method.* A symmetric two-step density control governs 10,000 steps of the two-step method. Step size sequence (top left) and energy error (bottom left) are plotted vs time. The step size varies by almost one order of magnitude, while energy is nearly preserved, with a maximum energy error of approximately 10^{-3} ; both step size and energy error are stable. The right graph shows the orbit. Precession and accuracy are comparable to those of Figure 2.

The Störmer Verlet method (explicit midpoint method) is a two-step method for $\ddot{q} = f(q)$. If combined with the two-step controller (24) it could therefore be compared to the one-step implementation and control of the previous section. Thus, in a simple numerical experiment, we solve the Kepler problem once more for eccentricity 0.8 with the two-step approach. The generated stepsize sequence, the energy error, as well as the trajectory are shown in Figure 3. The performance is similar to that of the adaptive one-step method.

General remarks. For $m \geq 2$, (22) typically includes a digital filter; by *averaging* $G(y_n)$ and $G(y_{n-1})$ it blocks $(-1)^n$ oscillations no matter what their source is. For $m = 3$, still within symmetric, reversible control, there is a choice between a 2nd order quadrature formula *with* a filter, and a 4th order formula *without* a filter. The latter requires weights in the quadrature formula (22) that preclude averaging; filtering and high quadrature order are conflicting objectives and can in general not be combined.

Although much more is known about the behavior of reversible one-step methods, the tests above indicate that step density control works well also for symmetric multistep methods applied to reversible problems. However, it remains an open question how to find the most reliable and suitable controllers for each problem and method class.

5 Adaptive multistep methods: 0–stability in the BDF method

In case the problem is dissipative and there is no need to preserve invariants, one can consider quadrature formulas with nonsymmetric weights. This implies that there are more controllers to choose from. In principle, one can consider both step density controllers and more conventional filter structures for generating the step size h directly. Here we shall focus on the latter approach.

A specification of how to implement digital filter step size generating algorithms was given in [19]. This may need several extensions, e.g., to include order control. However, instead of going into such details, we shall have a look at the basic problem of ensuring that the resulting method is “0–stable.”

The problem of variable step size 0–stability has received a considerable attention. For the particular case of the BDF methods, the best results known to date are those of Butcher and Heard [1]. The classical investigation finds the stability bounds on the step size ratios. This is an “open loop” approach; it disregards that the error estimate is fed back to a controller which changes the step size and interacts with the discretization method in a “closed loop.” As a controller can often stabilize an unstable process, stability is a property of the combination of method and controller, not a property of the method alone.

Here we shall test an implementation of the elementary controller and the digital filter *H211PI*, based on the specification in [19]. The discretization method is a fully variable coefficient BDF3, with error estimate provided by BDF2. The problem we consider is $\dot{y} = 0$. Each time this right-hand side is evaluated, we add a random perturbation $\{u_n\}$ of rectangular distribution on $[-10^{-4}, 10^{-4}]$. This creates a discrete process, where the random perturbation “excites” the controller by feeding it a non-zero error estimate. As the perturbation is small, the step size controller will attempt to increase the step size, typically to the extent that step size ratio limits are exceeded. These conditions are outside the assumptions behind the standard step size – error relation and are therefore especially challenging for the controller.

The limiter employed is of the form

$$\hat{x} \leftarrow 1 + \kappa \cdot \arctan((x - 1)/\kappa),$$

see [19]. It is smooth and avoids discontinuities introduced by logic and max or min functions. The parameter κ affects how strong the limiting action is; in the experiments below we will use $\kappa = 2$. The variable x represents either the ratio $\theta \cdot TOL/r_n$ (the control error), or the step size ratio, h_{n+1}/h_n . If x is large, then the limited value \hat{x} nevertheless does not exceed $1 + \kappa\pi/2$. Likewise,

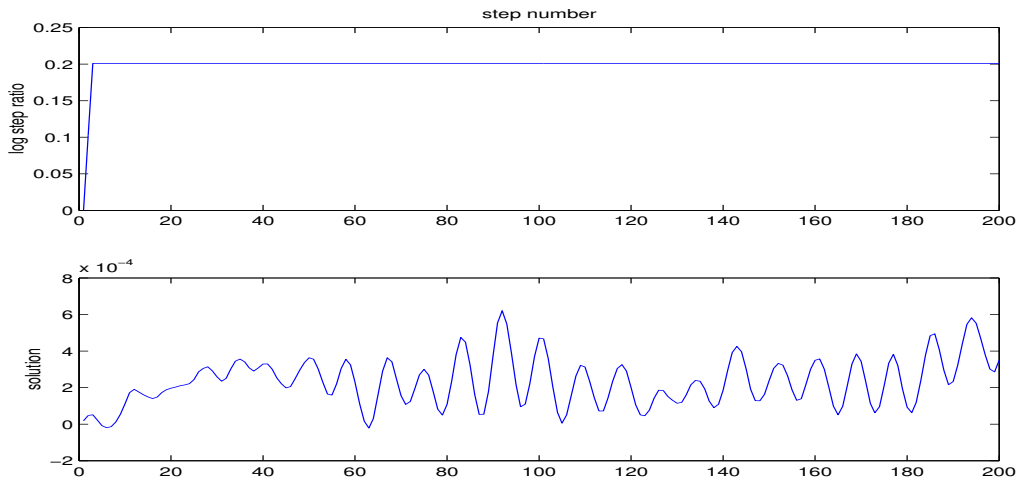


Fig. 4. *H211PI controller with dual limiters*. This simulation shows the logarithm of the step size ratio (top) and the numerical solution (bottom) vs step number. The controller “locks” at the maximum stable step size ratio ($10^{0.2} \approx 1.6$, [1]). An oscillation with an approximate period of 8.5 steps is clearly seen in the solution, indicating a frequency response peak (“resonance”) in the feedback control system. As the solution remains small ($\sim 10^{-4}$), the oscillation appears to have no serious consequences, although the solution does not represent a “random walk” process.

even if x is near 0, the value of \hat{x} remains finite. In normal operation, both the control error and the step size ratio should be near 1. Then $\hat{x} \approx x$; the limiter leaves x unaffected.

In the actual implementation of the controllers, be it the elementary controller (6) or a digital filter such as *H211PI*, (18), we choose to either omit the limiter completely, or to apply the limiter both to the control error and to the calculated step size ratio. Employing the limiter means that we first calculate the control error, then apply the limiter to prevent a large or small error to have excessive effects. Then the new step size is computed, and the limiter is applied again to make sure that the step ratio remains within bounds.

We have tested a large number of controllers and settings for the limiter when the BDF3 method is used. Only a few interesting results can be reported here. In Figure 4 we see how the *H211PI* filter performs when the limiter is applied both to control error and to step size ratios. The step size ratio would normally always exceed the maximum stable ratio of 1.618. As the limiter would also allow such a ratio and only removes excessive step size increases, the test in Figure 4 reveals that *the combination of BDF3 and H211PI automatically finds the maximum stable step size increase and stays on that limit without oscillations*.

This is partly akin to what happens when a suitable PI controller is used in combination with an explicit one-step method that encounters a mildly stiff

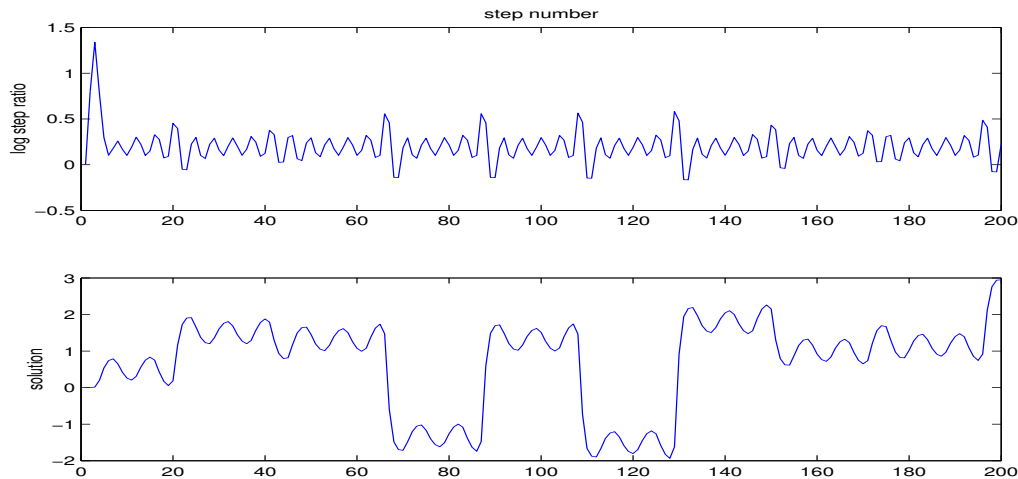


Fig. 5. *H211PI controller with limiters removed*. In this simulation, no restrictions are put on the control error or on the step size ratios. The step size ratios now oscillate around the maximum stable increase ratio, indicating stability problems. The solution no longer remains small, but is of order 1. The same periodicity as before is observed, although abrupt changes in solution magnitude now occur when the step ratio clearly exceeds the maximum stable ratio. Although overall performance is unsatisfactory, the feedback control system keeps exponential error growth at bay.

problem, [4]. The step size increases to the maximum stable step size, and stays at that limit without oscillations. However, we also observe that the numerical solution does not look like the expected “random walk” that would result from integrating the forcing function.

By removing the limiter, the controller can, if the perturbations push such changes, increase the step size freely. This will occasionally happen, as seen in Figure 5. For very large step size ratios, there are sudden abrupt changes in solution magnitude. The solution no longer remains small, the step ratios oscillate, and *there are clear indications of stability problems*. Thus we conclude that a limiter is necessary and increases robustness. The controller is not designed or analyzed for situations when the dynamics are far from those given by the asymptotic step size – error relation.

Figure 6 shows the performance of the elementary controller with limiters. The test is equivalent to that of Figure 4 as the setup is identical, save for the choice of controller. However, in this test we see that the elementary controller is unsuitable and much inferior to the *H211PI* in its interaction with BDF3.

The conclusion is that *the choice of controller is very important as are the precise implementation details*. Much remains to be done about the analysis of the interaction of method and controller, but it is already known that a careful implementation brings adaptive code performance much closer to the method’s theoretical properties, [19].

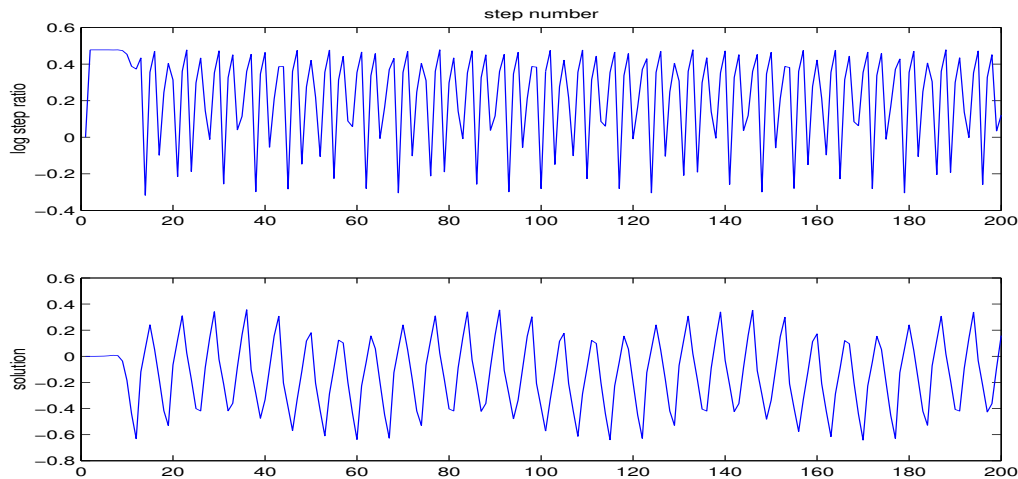


Fig. 6. *Elementary controller with dual limiters.* The elementary controller shows severe oscillations around the maximum stable step size ratio. In spite of employing the limiter, it is unable to prevent step size oscillations, and the numerical solution itself exhibits both a jagged oscillatory behavior as well as a sub-resonant periodicity of some 60 steps. The solution is of order 1.

6 Open problems

The problem of how to select the step size adaptively is fairly well understood for initial value ODE problems, but in most other areas, it appears that adaptive algorithms face both interesting and nontrivial problems. The recent developments clearly indicate that adaptivity can be significantly improved by a rigorous analysis, be it based on control theory, signal processing or the theory of systems with invariants. Step size selection algorithms benefit from being constructed with the same attention to analytic properties as the discretization methods themselves.

Multistep methods. Adaptive multistep methods have been used for initial value problems for a long time. Yet many problems remain, as has been indicated above. On the theoretical side, we have the problems of non-uniqueness of method representation, and step size sequence restrictions caused by the requirement of 0-stability. On the practical side, there is a need to find efficient adaptive algorithms that are both robust and amenable to analysis, bringing the actual implementation closer to theoretical performance.

Differential-algebraic equations. In initial value DAEs, the problem of adaptive step size selection is similar to that of ODEs. However, while in the ODE case the method often operates in the asymptotic regime, in DAEs some variables may obey the classical asymptotic model, while others behave differently. A controller designed to work well for the “classical” step size – error model $r = \varphi h^k$ may therefore not work for high index DAEs. The problem is to

identify the adequate step size – error model, or alternatively put, to design error estimators that have a proper behavior as a function of h .

If the error estimate in some ranges of h is *independent* of h , as may happen with some methods/error estimators, then that method/error estimator is useless and should be discarded. In adaptive time-stepping it is necessary to find combinations of methods and error estimators such that an increased step size causes an increased error (and vice versa). This is not only a matter of method and error estimator construction, but also a matter of the target problem class.

Geometric integration. In problems with invariants, it appears that step density control has opened up new possibilities. It is possible to vary the step size and increase both accuracy and efficiency while invariants are (nearly) preserved. However, there is still a need to investigate how accurately the step density control system needs to be integrated, and also whether phases errors can be decreased by more accurate quadrature rules for original time recovery.

In addition, much remains to make the technique robust in complex, large scale applications such as molecular dynamics. It also remains to be seen what the actual computational impact is in such cases.

Delay differential equations. Although much progress has been made in the numerical treatment of DDEs, this problem class has some distinctive features that make it especially hard to select time steps adaptively. In particular, DDEs typically exhibit discontinuities. The discontinuities have to be located, and the integration restarted from there. In problems with true delays only, the effect of discontinuities is not severe as the solution gets more and more regular with time. In “neutral” problems, however, discontinuities remain, and may recur very frequently if they are state dependent. In such problems one may be forced to seek alternative ways to adaptive step size selection than those presented here.

Stochastic differential equations. In SDEs one has a choice of simulating individual trajectories or computing moments that characterize the stochastic process. In the first case, adaptive step size selection may be of importance. However, when simulating realizations, the step size sequence cannot be chosen without great care. For example, power spectra must be correctly represented, and as these depend on the step size sequence, there are restrictions on step size selection algorithms.

Two-point boundary value problems. In 2p-BVPs, adaptivity is a matter of where to put grid points as well as grid refinement. Although this is not a “time step” selection process, it should be possible to use techniques based on digital filters. These need no longer be causal for selecting a *spatial* discretization. An approach combining such filtering with adaptive step density selection

has to this author's knowledge not been tried for 2p-BVPs, but might be advantageous.

Partial differential equations. In initial-boundary value problems adaptivity is a very challenging problem. Parabolic and hyperbolic problems show rather different behaviors. In the former case, the ODE techniques may very well be quite successful, provided that the spatial discretization can be kept fixed. In hyperbolic problems, on the other hand, there is often a strong space–time interaction; a wave may propagate in a certain direction in space and with a specific speed, and it may be desirable to track a wave-front using a locally fine grid. This implies that one needs to combine adaptive ODE-IVP techniques with adaptive 2p-BVP techniques, perhaps taking physics into account. Although such moving grids have been used in practice, much remains in understanding how such techniques should be designed and how they affect the numerical solution.

References

- [1] J.C. BUTCHER AND A.D. HEARD. Stability of numerical methods for ordinary differential equations. *Numerical Algorithms* 31:59–73, 2002.
- [2] C.W. GEAR. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice–Hall, Englewood Cliffs 1971.
- [3] K. GUSTAFSSON, M. LUNDH AND G. SÖDERLIND. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT* 28:270–287, 1988.
- [4] K. GUSTAFSSON. Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods. *ACM TOMS* 17:533–554, 1991.
- [5] K. GUSTAFSSON. Control theoretic techniques for stepsize selection in implicit Runge–Kutta methods. *ACM TOMS* 20:496–517, 1994.
- [6] K. GUSTAFSSON AND G. SÖDERLIND. Control strategies for the iterative solution of nonlinear equations in ODE solvers. *SIAM J. Sci. Comp.* 18:23–40, 1997.
- [7] E. HAIRER AND G. WANNER. *Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems*. Springer-Verlag, 2nd revised edition, Berlin 1996.
- [8] E. HAIRER, CH. LUBICH AND G. WANNER. *Geometric Numerical Integration*. Springer-Verlag, Berlin 2002.
- [9] E. HAIRER AND G. SÖDERLIND. Explicit, time reversible, adaptive step size control. Accepted for publication in *SIAM J. Sci. Comput.*, 2005.
- [10] G. HALL. Equilibrium states of Runge–Kutta schemes. *ACM TOMS* 11:289–301, 1985.

- [11] G. HALL. Equilibrium states of Runge–Kutta schemes, part II. *ACM TOMS* 12:183–192, 1986.
- [12] G. HALL AND D. HIGHAM. Analysis of stepsize selection schemes for Runge–Kutta codes. *IMA J. Num. Anal.* 8:305–310, 1988.
- [13] D. HIGHAM AND G. HALL. Embedded Runge–Kutta formulae with stable equilibrium states. *J. Comp. and Appl. Math.* 29:25–33, 1990.
- [14] H. OLSSON AND G. SÖDERLIND. Stage value predictors and efficient Newton iterations in implicit Runge–Kutta methods. *SIAM J. Sci. Comput.* 20:185–202, 1999.
- [15] H. OLSSON AND G. SÖDERLIND. The approximate Runge–Kutta computational process. *BIT* 40:351–373, 2000.
- [16] J.J.B. DE SWART AND G. SÖDERLIND. On the construction of error estimators for implicit Runge–Kutta methods. *J. Comp. and Appl. Math.* 86:347–358, 1997
- [17] G. SÖDERLIND. Automatic control and adaptive time–stepping. *Numerical Algorithms* 31:281–310, 2002.
- [18] G. SÖDERLIND. Digital filters in adaptive time–stepping. *ACM Trans. Math. Software*, 29:1–26, 2003.
- [19] G. SÖDERLIND AND L. WANG. Adaptive time–stepping and computational stability. To appear in *J. Comp. and Appl. Math.*, 2005.
- [20] G. SÖDERLIND AND L. WANG. Evaluating Numerical ODE/DAE Methods, Algorithms and Software. To appear in *J. Comp. and Appl. Math.*, 2005.
- [21] D. STOFFER. Variable steps for reversible integration methods. *Computing* 55:1–22, 1995.
- [22] H.A. WATTS. Step size control in ordinary differential equation solvers. *Trans. Soc. Comput. Sim.* 1:15–25, 1984
- [23] J.A. ZONNEVELD. Automatic numerical integration. Ph.D. thesis, Math. Centre Tracts 8, CWI, Amsterdam 1964.