Numerical Algorithms (2021) 86:537–563
https://doi.org/10.1007/s11075-020-00900-1

ORIGINAL PAPER

Local error estimation and step size control in adaptive linear multistep methods

Carmen Arévalo[1] · Gustaf Söderlind[1] · Yiannis Hadjimichael[2] · Imre Fekete[2,3]

Received: 11 June 2019 / Accepted: 30 January 2020 / Published online: 5 June 2020

Abstract

In a k-step adaptive linear multistep methods the coefficients depend on the $k-1$ most recent step size ratios. In a similar way, both the actual and the estimated local error will depend on these step ratios. The classical error model has been the asymptotic model, $ch^{p+1}y^{(p+1)}(t)$, based on the constant step size analysis, where all past step sizes simultaneously go to zero. This does not reflect actual computations with multistep methods, where the step size control selects the next step, based on error information from previously accepted steps and the recent step size history. In variable step size implementations the error model must therefore be dynamic and include past step ratios, even in the asymptotic regime. In this paper we derive dynamic asymptotic models of the local error and its estimator, and show how to use dynamically compensated step size controllers that keep the asymptotic local error near a prescribed tolerance TOL. The new error models enable the use of controllers with enhanced stability, producing more regular step size sequences. Numerical examples illustrate the impact of dynamically compensated control, and that the proper choice of error estimator affects efficiency.

Keywords Adaptivity · Time stepping · Step size control · Local error estimation · Linear multistep methods · Dynamic compensator · Variable step size · Differential equations · Initial value problems · Control theory

1 Introduction

We shall consider a standard initial value problem

$$\dot{y} = f(t, y), \qquad y(0) = y_0,$$

✉ Gustaf Söderlind
Gustaf.Soderlind@gmail.com

Extended author information available on the last page of the article.

and let y_n denote the numerical approximation to $y(t_n)$. Further, we let y'_n denote $f(t_n, y_n)$. Thus derivatives of functions are denoted by a dot, while samples of the vector field are denoted by prime. The methodology developed here does not depend on whether the problem is scalar or a system of equations.

We will study linear multistep methods on nonuniform grids for this initial value problem, of the *normalized* form

$$y_{n+1} + \sum_{j=1}^{k} a_{k-j,n} y_{n-j+1} = \sum_{j=0}^{k} b_{k-j,n}\, h_n y'_{n-j+1}, \tag{1}$$

where $h_n = t_{n+1} - t_n$. Normalization will also be discussed in connection with error constants. The subscript n of the coefficients indicates that these are rational functions of the *step ratios*, $\rho_{n-1} = h_n/h_{n-1}$. The method is implicit if $b_{k,n} \neq 0$. We shall consider the grid-independent representation, [1], where every linear multistep method is defined in terms of a fixed set of parameters and interpolation conditions. Given any grid point distribution $\{t_n\}_0^\infty$, this representation generates the coefficients $a_{k-j,n}$ and $b_{k-j,n}$.

Time-step adaptivity is of key importance to make initial value problem solvers efficient. In particular, in stiff problems, where time scales may vary by several orders of magnitude, it is necessary to vary the step size accordingly. If possible, an implementation should offer *tolerance proportionality*, i.e., the global error should be proportional to a preset accuracy requirement, TOL.

The classical approach has been to assume that the local error can be represented asymptotically by

$$r_n = \varphi_n h_n^q, \tag{2}$$

where φ_n is the norm of the principal error function, h_n is the current time step, and r_n is the norm of the error estimate. The time step is then typically changed according to the elementary control law

$$h_{n+1} = \left(\frac{\mathrm{TOL}}{r_n}\right)^{1/q} h_n. \tag{3}$$

For a method of order p, we take $q = p+1$ or $q = p$ depending on whether the local error per step (EPS) or the local error per unit step (EPUS) is controlled. The scheme is usually accompanied by safety heuristics, and detailed descriptions can e.g. be found in monographs such as [3, 6, 12, 13, 17].

Starting in the late 80s, adaptivity based on control theory was developed, [9–11], leading to [18] and a framework for using digital filters and signal processing, [19, 21]. Computational stability was further studied in [20]. Special controllers for explicit and implicit geometric integration were also developed in [14] and [23]. Combinations of these techniques have been applied to near-Hamiltonian problems with weak Rayleigh damping, [16].

This theory has focused on one-step methods, for which the asymptotic error model (2) applies whenever the tolerance is small enough. Taking logarithms, (2) becomes

$$\log r_n = q \cdot \log h_n + \log \varphi_n, \tag{4}$$

suggesting that a step size change affects the error (approximately) by a constant "gain" factor of q, without any dependence on step size history. While this static error model is correct for one-step methods in the asymptotic regime, it is no longer correct outside asymptotic operation, [10].

For a k-step linear multistep method it is not even correct in the asymptotic regime. As the method coefficients depend on $k-1$ past step ratios, errors are similarly affected, and the error's step size dependence becomes dynamic. Nevertheless, the elementary controller (3) has been applied to multistep methods, with varying success. Unfortunately, its stability may deteriorate in combination with e.g. the Adams–Bashforth methods. Thus, depending on the choice of error estimator, the dynamic interaction of method and controller may become resonant and oscillatory. The problems can be overcome by a careful combination of method, estimator and controller.

In order to improve adaptive multistep methods, we shall investigate new *dynamic error models* of *multiplicative form*,

$$r_n = \varphi_n h_n^q \cdot \prod_{j=1}^{s} \rho_{n-j}^{\delta_j}. \tag{5}$$

Here the first factor is identical to the static model (2). The second factor accounts for step size history in terms of step ratios $\rho_{n-j} = h_{n-j+1}/h_{n-j}$. The number of past step ratios is $s = k-1$ for a k-step method if r_n represents the actual error, while it is $s = k$ for error estimators that use additional data from one previous step. The parameters δ_j are characteristic of each multistep method and its error estimator.

In multistep methods, there is a concern over zero stability when the step size varies, focusing on how large step ratios can be allowed without causing instability, see e.g. [5, 7, 8, 22]. While this issue is of significance, it does not reflect that feedback control interacts with the method and manages overall stability. Thus, in a strongly zero stable method a stable computational process can be maintained by generating a smooth step size sequence (i.e., having step ratios $\rho_n \approx 1$). This can be achieved by a well designed controller, working with incremental step size changes, avoiding traditional step size halving/doubling.

The grid-independent approach to multistep methods has also been implemented in a proof-of-concept software platform, [2], where any multistep method can be evaluated under *ceteris paribus* conditions. While the theory makes it possible to find method-specific dynamic error estimators in terms of current and past step sizes, the implementation has so far only offered controllers for the static model (2). Our main objectives in this paper are therefore **1)** *to analyze dynamic error estimators*, based on models of the form (5); and **2)** *to devise suitable controllers* that manage the local error as well as the stability of the interaction of method and controller. Our approach is based on using discrete control theory as outlined in [19].

The main result is that we construct a dynamic compensator that extracts the static part of the asymptotic error model from the estimator, allowing us to employ standard digital filters and other controllers to achieve results comparable to those of one-step methods. We further show that without this dynamic compensation, conventional techniques are not completely reliable. Thus, for example, the elementary controller

(3) is unsuitable for the control of adaptive linear multistep methods, as the stability margin of the process deteriorates with increasing step size history dependence.

2 Dynamic error models and control objectives

In the classical constant step size theory, the local error of a multistep method has an asymptotic expansion with leading term

$$l_n = c^* h^{p+1} y^{(p+1)}(t_n), \tag{6}$$

as $h \to 0$, where c^* is referred to as the method's (normalized) *error constant*. But this assumes that all steps shrink simultaneously. In adaptive methods, however, having reached the point $t_{n+1} = t_n + h_n$, we consider changing the next step size, h_{n+1}, but none of the previously accepted steps $h_n, \ldots, h_{n-k+1}$. As a consequence, the asymptotic error will depend on previous step ratios.

Local error control is justified by the possibility of linking a local error bound to the global error via tolerance proportionality. Global error accumulation can be modeled by the variational equation, $\dot{u} = J(t)u + w$, where $J(t) = \partial f/\partial y$ along the exact solution $y(t)$. Here $u(t)$ and $w(t)$ represent global and local errors, respectively. Let $\mu[J(t)]$ denote the logarithmic norm of $J(t)$, and assume that $\mu[J(t)] \leq M$ for all $t \geq 0$. Note that M may be negative. Further, let $\|w\|_\infty = \max_{t\geq 0} \|w\|$. Then a global error bound can be obtained by integrating the differential inequality

$$\frac{\mathrm{d}}{\mathrm{dt}}\|u\| \leq M \cdot \|u\| + \|w\|_\infty$$

over a single step to obtain

$$\|u(t+h)\| \leq \mathrm{e}^{hM}\|u(t)\| + \frac{\mathrm{e}^{hM} - 1}{M}\|w\|_\infty. \tag{7}$$

In *nonstiff computation*, we have $|hM| \ll 1$. Apart from the error propagation, the global error is then incremented by approximately $h\|w\|_\infty$ in a step of length h. By using the control objective $\|w\|_\infty \leq \text{TOL}$ we keep the global error growth rate in check and achieve tolerance proportionality. This corresponds to local error per unit step (EPUS) control.

If on the other hand $-hM \gg 1$, modeling *stiff computation* and strong damping, the exponential terms in (7) can be neglected, and we have

$$\|u(t+h)\| \lesssim -\frac{\|w\|_\infty}{M} = -\frac{h\|w\|_\infty}{hM} \ll h\|w\|_\infty.$$

Thus, with little or no error propagation, the global error effectively equals (a fraction of) the local error. To achieve tolerance proportionality, it is then sufficient to use the control objective $h\|w\|_\infty \leq \text{TOL}$, corresponding to local error per step (EPS) control. The choice of EPUS or EPS determines the power q in the error model, but by suitably expressing the controller's parameters in terms of q, one can obtain similar overall dynamics for both control objectives as well as for methods of different orders.

Inserting the exact solution $y(t)$ into (1), we obtain a residual,

$$y(t_{n+1}) + \sum_{j=1}^{k} a_{k-j,n} y(t_{n-j+1}) - \sum_{j=0}^{k} b_{k-j,n}\, h_n \dot{y}(t_{n-j+1}) = -\tilde{l}_n, \tag{8}$$

where $\tilde{l}_n = c(\bar{\rho}_n)\, h_n^{p+1} y^{(p+1)}(t_n)$. Compared to (6), the error constant c^* has been replaced by a function $c(\bar{\rho}_n)$, which depends on the $k-1$ most recent step ratios, collected in the vector $\bar{\rho}_n = (\rho_{n-1}, \ldots, \rho_{n-k+1})$. Letting $u_n = y_n - y(t_n)$ denote the global error, we subtract (8) from (1) and linearize to obtain

$$u_{n+1} + \sum_{j=1}^{k} a_{k-j,n} u_{n-j+1} - \sum_{j=0}^{k} b_{k-j,n}\, h_n [J(t_{n-j+1}) u_{n-j+1} + \frac{\tilde{l}_n}{b(\bar{\rho}_n) h_n}] = 0,$$

where $b(\bar{\rho}_n) = \sum_0^k b_{k-j,n}$. This is the multistep discretization of the variational equation $\dot{u} = J(t) \cdot u + w$. The function w is constant within each step, and

$$w = \frac{\tilde{l}_n}{b(\bar{\rho}_n) h_n} = y^{(p+1)}(t_n)\, h_n^p \cdot \pi(\bar{\rho}_n), \tag{9}$$

where $\pi(\bar{\rho}_n) = c(\bar{\rho}_n)/b(\bar{\rho}_n)$ introduces dynamics into the error model.

In constant step size theory, $\bar{\rho} = (1, 1, \ldots, 1) = \mathbf{1}$. The method is often normalized so that $b(\mathbf{1}) = 1$, implying that $c(\mathbf{1}) = c^*$, cf. [12, p. 373]. It then follows that $\pi(\mathbf{1}) = c^*$. Since we have used the normalization $a_{k,n} = 1$ instead, the error constant is affected. Let $\hat{\pi}(\bar{\rho}) = \pi(\bar{\rho})/\pi(\mathbf{1})$. Then (9) becomes

$$w = \hat{c}^*\, y^{(p+1)}(t_n)\, h_n^p \cdot \hat{\pi}(\bar{\rho}_n), \tag{10}$$

making the error constant $\hat{c}^* = \pi(\mathbf{1})$ conform to the method normalization and error propagation model above. For the Adams methods, it always holds that $b(\bar{\rho}_n) \equiv 1$, implying that $\hat{c}^* = c^*$ even when our normalization is used.

For general multistep methods, we have thus arrived at the error model

$$r_n = \varphi_n h_n^q \cdot \hat{\pi}(\bar{\rho}_n), \tag{11}$$

with $q = p$ for EPUS control and $q = p + 1$ for EPS. Let $\bar{\delta} = \{\delta_j\}_1^s$ be given by the gradient

$$\bar{\delta}^{\mathrm{T}} = \operatorname{grad}_{\bar{\rho}} \hat{\pi}(\bar{\rho})\big|_{\bar{\rho}=\mathbf{1}}. \tag{12}$$

Assume that $\bar{\rho}_n = \mathbf{1} + \bar{v}_n$ with $\|\bar{v}_n\|_\infty \ll 1$. It follows that $\log \bar{\rho}_n \approx \bar{v}_n$, and

$$\log \hat{\pi}(\bar{\rho}_n) \approx \log \Big[1 + \bar{\delta}^{\mathrm{T}} \cdot \bar{v}_n\Big] \approx \sum_{j=1}^{s} \delta_j v_{n-j} \approx \sum_{j=1}^{s} \log \rho_{n-j}^{\delta_j} = \log \prod_{j=1}^{s} \rho_{n-j}^{\delta_j}.$$

Hence the approximation $\hat{\pi}(\bar{\rho}_n) \approx \prod_1^s \rho_{n-j}^{\delta_j}$ is accurate to $\mathrm{O}(\|\bar{v}\|_\infty^2)$, and establishes the multiplicative dynamic error model (5). This approximation enables the design of a controller that compensates the specific linear dynamics of the error. It is also a necessary simplification when the step number k is large, as the rational function $\hat{\pi}(\bar{\rho}_n)$ consists of multinomials with a very large number of terms. For this reason, the determination of the powers δ_j typically requires symbolic computation.

The error model is best illustrated by studying a simple example, and we choose the two-step Adams–Bashforth (AB2) of order $p = 2$. In the grid-independent formulation [1] the adaptive AB2 method reads

$$y_{n+1} = y_n + \left(1 + \frac{\rho_{n-1}}{2}\right) h_n y_n' - \frac{\rho_{n-1}}{2}\, h_n y_{n-1}', \tag{13}$$

where $h_n = t_{n+1} - t_n$. The local error $r_n = y(t_{n+1}) - y_{n+1}$ at t_{n+1} is found by inserting exact data, i.e., $y_n = y(t_n)$, $y_n' = \dot{y}(t_n)$ and $y_{n-1}' = \dot{y}(t_{n-1})$, and expanding in a Taylor series to obtain the asymptotic error expression

$$\tilde{l}_n \approx \frac{5}{12} \dddot{y}(t_{n+1}) h_n^3 \left(\frac{2}{5} + \frac{3}{5\rho_{n-1}}\right), \tag{14}$$

where higher order terms (here fourth order in the step size) have been neglected. For constant step size ($\rho_{n-1} = 1$) we recognize the classical asymptotic *principal error function* of the AB2 with its error constant, as $\varphi_n = 5\dddot{y}/12$.

For variable steps, we now have

$$\hat{\pi}(\rho) = \frac{2}{5} + \frac{3}{5\rho} \quad \Rightarrow \quad \delta_1 = \hat{\pi}'(1) = -\frac{3}{5}. \tag{15}$$

Consequently, the asymptotic local error model in multiplicative form (5) is

$$\tilde{l}_n \approx \varphi_n\, h_n^3 \cdot \rho_{n-1}^{-3/5} = \varphi_n\, h_n^{12/5}\, h_{n-1}^{3/5}. \tag{16}$$

While “dimensionless,” the step ratio does not affect the order q, but modifies individual powers of the step sizes. This has little influence on *error magnitude* (as $\rho_{n-1} \approx 1$) but alters *step size dynamics* and the subsequent control design.

In one-step methods, the elementary controller is derived from (2) by assuming that φ is slowly varying along the solution to the differential equation. Thus, if the error r_n deviates from TOL, the next step size can be chosen to make $r_{n+1} \approx \text{TOL}$ by solving

$$\text{TOL} = \varphi_{n+1} h_{n+1}^q, \tag{17}$$

for h_{n+1}, assuming that $\varphi_{n+1} = \varphi_n$. Dividing (17) by (2) we obtain the control law (3), known as a *deadbeat controller*, [19]. This has finite impulse response (FIR), since an impulse in the sequence $\{\varphi_j\}_0^\infty$ will be compensated by the controller in a finite number of steps, in this case a single step. In fact, it is easily shown that the control law yields $\varphi_n h_{n+1}^q \approx \text{TOL}$, i.e., the proposed step size is matched to the previous value of the principal error function.

The same intuitive approach could be considered for multistep methods. Assuming that $\varphi_{n+1} = \varphi_n$ in the AB2 method and dividing the two equations

$$\begin{aligned} \text{TOL} &= \varphi_{n+1}\, h_{n+1}^{12/5}\, h_n^{3/5} \\ r_n &= \varphi_n\, h_n^{12/5}\, h_{n-1}^{3/5} \end{aligned}$$

results in the control law

$$h_{n+1} = \left(\frac{\text{TOL}}{r_n}\right)^{5/12} \rho_{n-1}^{-1/4} \cdot h_n. \tag{18}$$

Apart from the unexpected power of TOL$/r_n$ in (18), the controller has an extra factor due to the error's dynamic character. Perhaps more surprisingly, (18) is not deadbeat, and has a less desirable behavior.

There are also alternative ways to estimate the error, and a few will be described in detail in Section 5. Above, the error (16) was obtained by Taylor series expansion. It can be computed in practice, using a comparison method (at least) one order higher than AB2, say the AB3, operating in the asymptotic regime. Another common estimation technique is to compare the results from two methods of the same order. This is similar to the classical predictor–corrector technique. Thus, if the AB2 method is represented as a method constructing a polynomial of degree 2, the error can be estimated by comparing this polynomial at t_{n+1} to the result obtained by extrapolating the corresponding polynomial, constructed on the previous step. While crude, this difference can still be used to compute an asymptotically correct error. The dynamic asymptotic error model is then more complex,

$$r_n = \varphi_n \, h_n^q \, \rho_{n-1}^{-45/23} \, \rho_{n-2}^{-12/23}. \tag{19}$$

An attempt to derive control laws by the simple approach used above fails, and results in unstable controllers, both in EPUS and EPS modes. The elementary control law (3) fares a little better, and is stable in EPS mode in combination with (19), but unstable in EPUS mode. Thus, if methods, estimators and controllers are combined arbitrarily, performance is often underwhelming, see Fig. 1, where the elementary controller is compared to one of the dedicated dynamically compensated controllers we will develop in this paper. This motivates a thorough investigation of error estimation and step size control for adaptive multistep methods, focusing on the dynamics of the error model.

3 Tools from linear control theory

In the implementation of a discrete time system, it is important to synchronize variable indexation. When using the z transform, all variables, sequences and transforms are expressed in relation to the reference index n, corresponding to the point t_n, and we use the following time domain numbering convention:

1. Given the numerical solution (t_n, y_n), the method takes a step size h_n to compute y_{n+1} at $t_{n+1} = t_n + h_n$
2. The method produces an error estimate r_n at t_{n+1}, modeled in the asymptotic regime by an error model such as (5)
3. The controller computes the next step size ratio ρ_n from r_n
4. The controller then updates the step size according to $h_{n+1} = \rho_n h_n$
5. The computational process repeats from 1.

We shall analyze this process using linear control theory, which is based on the analysis of linear difference equations. The multiplicative forms of the error model and controller are necessary, since they can be converted to difference equations by taking logarithms, after which the z transform is used to analyze stability and

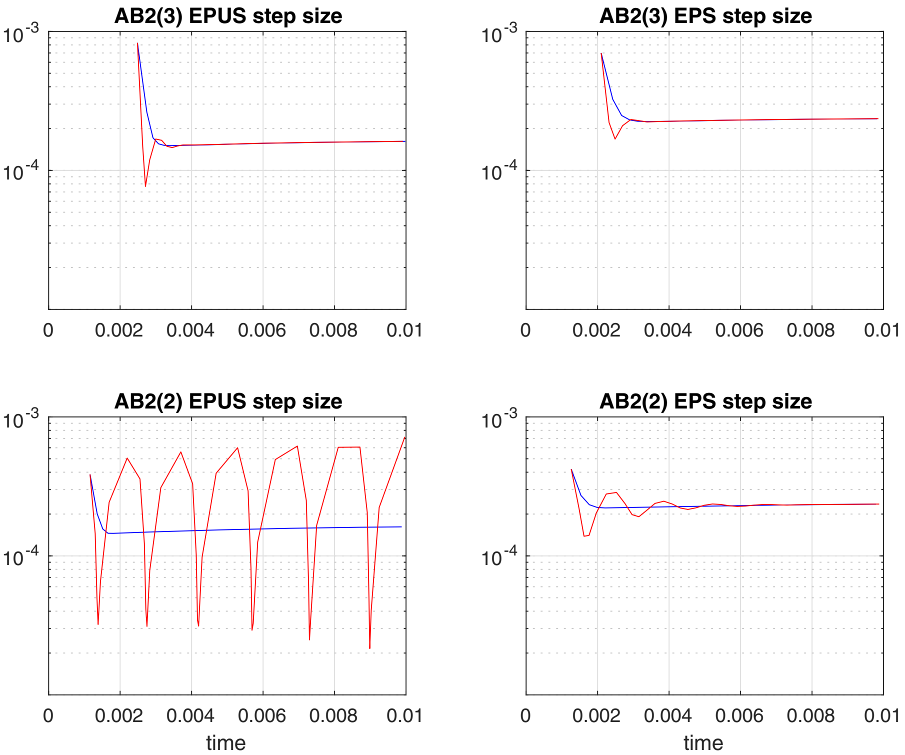

Fig. 1 The AB2 method is combined with two different error estimators and two different controllers, running in EPUS and EPS modes. Top panels show emulated step size sequences with a pre-constructed $\varphi(t)$ for AB2(3) with third order error estimator. Lower panels show corresponding data for AB2(2) with second order error estimator. The tolerances were $\mathrm{TOL} = 2 \cdot 10^{-6}$ (EPUS) and $\mathrm{TOL} = 10^{-9}$ (EPS), respectively. Oscillatory sequences (red) are due to resonances, caused by the elementary controller (3). Smooth sequences (blue) are generated by a dynamically compensated controller with vastly improved stability. Lower left panel shows that AB2(2) is unstable in EPUS mode when the elementary controller is used (red)

frequency response. For a simple demonstration of this procedure, let us consider (5) for $s = 2$, i.e.,

$$r_n = \varphi_n \, h_n^q \cdot \rho_{n-1}^{\delta_1} \, \rho_{n-2}^{\delta_2} = \varphi_n \, h_n^{q+\delta_1} \, h_{n-1}^{\delta_2-\delta_1} \, h_{n-2}^{-\delta_2}. \tag{20}$$

Now, taking logarithms in (20), we obtain the difference equation

$$\log r_n = G(E) \log h_n + \log \varphi_n, \tag{21}$$

where E is the *forward shift operator*, informally defined by $Eu_n = u_{n+1}$ for any discrete sequence. The rational function G is defined by

$$G(z) = q + \delta_1 + (\delta_2 - \delta_1)z^{-1} - \delta_2 z^{-2} = \frac{(q+\delta_1)z^2 + (\delta_2 - \delta_1)z - \delta_2}{z^2}. \tag{22}$$

Note that, due to the structure of (20), it follows that $G(1) = q$.

We let $\mathbf{h} = \{h_j\}_0^\infty$ denote the entire sequence of step sizes, and use the simplified notation $\log \mathbf{h} = \{\log h_j\}_0^\infty$. A standard tool in discrete control theory is the z *transform*, defined for the sequence $\log \mathbf{h}$ by

$$(\mathcal{Z} \log \mathbf{h})(z) = \sum_{j=0}^{\infty} (\log h_j) z^{-j}.$$

Since the risk of misunderstanding is minimal, with a slight abuse of notation we follow [19] and let $\log h$ denote the z transform of the sequence $\log \mathbf{h}$. The z transform of the linear difference equation (21) is then

$$\log r = G(z) \log h + \log \varphi. \tag{23}$$

The function $G(z)$, referred to as the *process model*, is the z transform of the operator associated with the *step size–error relation* (21), while the additive term $\log \varphi$ is an *external disturbance* corresponding to the principal error function, which is to be "rejected" by the controller. This means that the controller adjusts the step size to counteract the influence of $\log \varphi$.

In order to control the error magnitude by adjusting the step size, the error sequence $\log \mathbf{r}$ is compared to the tolerance $\log \text{TOL}$, and the sequence

$$\left\{ \log \frac{\text{TOL}}{r_j} \right\}_0^\infty = \log \text{TOL} - \log \mathbf{r}$$

is referred to as the *control error*. The controller aims to eliminate the control error, continually adjusting the step size according to the *control law*

$$\log h = C(z) \cdot (\log \text{TOL} - \log r), \tag{24}$$

where $C(z)$ is the z transform of the linear difference operator associated with the controller. The overall interaction between process and controller is usually represented by a block diagram, see Fig. 2.

Next we analyze the process–controller interaction. Combining the control law with the process model, we have the linear system

$$\begin{aligned} \log h + C(z) \log r &= C(z) \log \text{TOL} \\ -G(z) \log h + \log r &= \log \varphi. \end{aligned}$$

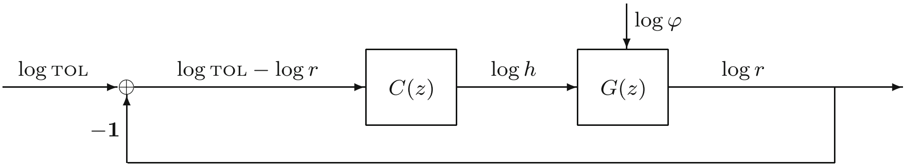

Fig. 2 **Time step adaptivity viewed as a feedback control system**. Represented by the transfer function $G(z)$, the computational process takes $\log h$ as input, producing an error estimate $\log r = G(z) \log h + \log \varphi$. The principal error function $\log \varphi$ enters as an additive disturbance, to be compensated by the controller. The error estimate $\log r$ is fed back and compared to $\log \text{TOL}$. The controller, represented by its transfer function $C(z)$, constructs the next step size through $\log h = C(z) \cdot (\log \text{TOL} - \log r)$. All sequences, as well as the controller and the process, are represented by their z transforms

Solving for $\log h$ and $\log r$ in terms of the data $\log \varphi$ and $\log \mathrm{TOL}$, we obtain

$$\begin{aligned}\log h &= H_{\varphi}(z) \log \varphi + H_{\mathrm{TOL}}(z) \log \mathrm{TOL} \\ \log r &= R_{\varphi}(z) \log \varphi + R_{\mathrm{TOL}}(z) \log \mathrm{TOL},\end{aligned}$$

where we identify the four *closed loop transfer functions*, from $\log \varphi$ and $\log \mathrm{TOL}$ respectively, to $\log h$ and $\log r$. As the constant TOL is of little interest, we take $\mathrm{TOL} = 1$ ($\log \mathrm{TOL} := 0$), meaning that the error is measured in units of TOL. Note that changing the constant TOL has no effect on the error, since $R_{\mathrm{TOL}}(z) \log \mathrm{TOL} = 0$ for any constant TOL. Likewise, it has little effect on the step size, other than adding a constant to the step size sequence $\log \mathbf{h}$.

The *step size transfer function* and *error transfer function* are, respectively,

$$H_{\varphi}(z) = -\frac{C(z)}{1 + C(z)G(z)}; \qquad R_{\varphi}(z) = \frac{1}{1 + C(z)G(z)}. \tag{25}$$

The asymptotic process model $G(z)$ is found by analyzing the error estimator of a given linear multistep method, while the controller $C(z)$ can be chosen to meet various design criteria for stability and smooth step size sequences.

The closed loop transfer functions have the same poles, which correspond to the zeros of the characteristic equation and govern *stability*. They must remain well located inside the unit circle. It is sometimes possible to construct $C(z)$ so as to move all poles to $z = 0$, corresponding to *deadbeat control*.

To model a bounded, quasi-periodic input one frequency at a time, we take $\log \varphi = \{\mathrm{e}^{\mathrm{i}\omega n}\}$ for $\omega \in [0, \pi]$, with $\omega = \pi$ corresponding to the highest discrete frequency, $\log \varphi_n = (-1)^n$. For such periodic inputs, it holds that

$$\log h = H_{\varphi}(\mathrm{e}^{\mathrm{i}\omega}) \log \varphi; \qquad \log r = R_{\varphi}(\mathrm{e}^{\mathrm{i}\omega}) \log \varphi$$

where the complex value $H_{\varphi}(\mathrm{e}^{\mathrm{i}\omega})$ reveals the phase and amplitude of the step size sequence compared to the input $\log \varphi = \{\mathrm{e}^{\mathrm{i}\omega n}\}$. Disregarding phase, the *scaled step size frequency response* and the *error frequency response* are given by, respectively,

$$A_h(\omega) = |q H_{\varphi}(\mathrm{e}^{\mathrm{i}\omega})|; \qquad A_r(\omega) = |R_{\varphi}(\mathrm{e}^{\mathrm{i}\omega})|, \tag{26}$$

where the extra factor of q in $|q H_{\varphi}(\mathrm{e}^{\mathrm{i}\omega})|$ normalizes the step size attenuation so that $A_h(0) = 1$. This factor is only a matter of convenience, reflecting that for the asymptotic process it always holds that $G(1) = q$.

Step size and error attenuations are dimensionless ratios, measured in the ISO unit of decibel (dB), defined by $20 \cdot \log_{10} A_h(\omega)$. Thus an attenuation of $+20$ dB corresponds to an amplification by one order of magnitude. Likewise, an attenuation of -6 dB corresponds to a damping by a factor of 2.

The zeros (if any) of the step size transfer function $|q H_{\varphi}(z)|$ *block signal transmission*. Thus, selecting $C(z)$ such that $A_h(\pi) = 0$, one can design low-pass digital filters that suppress noise and produce smooth step size sequences. This is covered in detail for the static process $G(z) = q$ in [19, 21].

In Fig. 3 we plot the step size frequency responses corresponding to the AB2 method operating in EPUS and EPS modes, using two different error estimators and the same two controllers as in Fig. 1, with a one-to-one correspondence between the plots. As before, the designation AB2(3) refers to a third order error estimator,

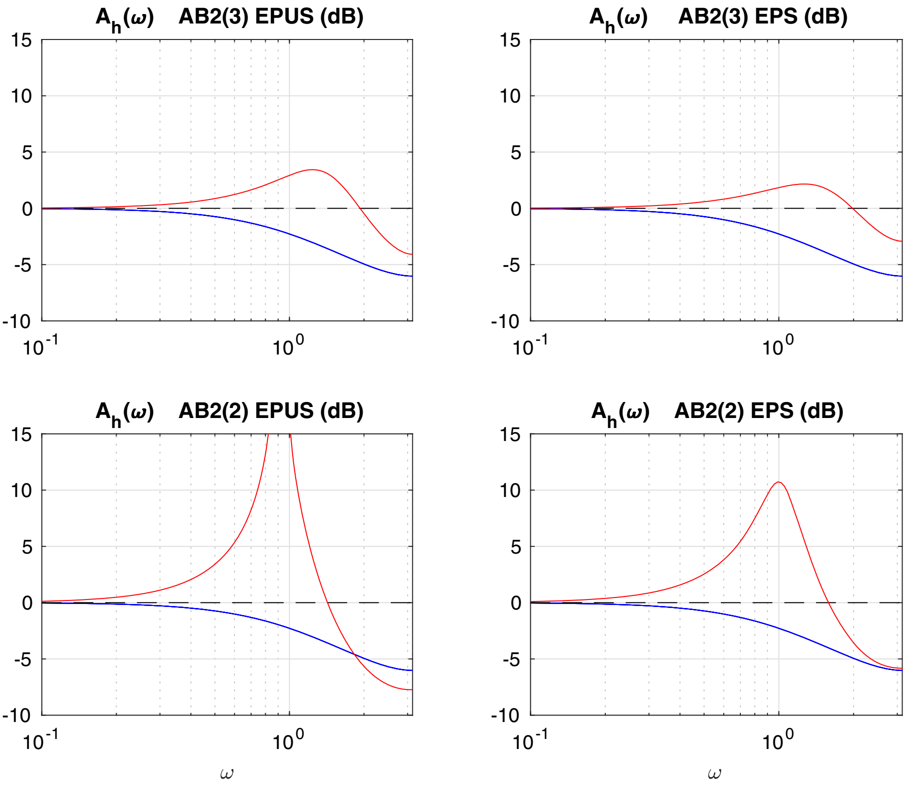

Fig. 3 Frequency response diagrams show step size attenuation when the AB2 method is combined with two different controllers, and two different error estimators in EPUS mode (left panels) and EPS mode (right panels). The elementary controller (red) has strong resonances near $\omega \approx 1$, making it unsuitable for multistep methods. With the third order estimator (top panels) the resonance is less pronounced. By contrast, a dynamically compensated controller (blue) eliminates the resonance and offers additional high frequency suppression, generating smooth step size sequences. Dashed black reference curve represents deadbeat control. Attenuation is measured in decibels, dB

while AB2(2) refers to a second order estimator. For the latter, the frequency response reveals that the elementary controller has a damaging resonance near $\omega \approx 1$. This contaminates the transient performance of the elementary controller, even causing the unstable oscillatory behavior in EPUS mode observed in Fig. 1. Although EPUS mode is always worse than EPS, the resonance is much less prominent for the third order estimator. There are two conclusions to be drawn. First, as performance differs strongly, it is important to combine good error estimators with controllers so as to avoid resonance and instability. While the choice of error estimator is important, a dynamically compensated controller is always better than the elementary controller, which is *unsuitable for multistep methods*. Figure 3 demonstrates that a dynamically compensated controller cannot only manage the benign third order estimator, but also the second order estimator, with a nearly uniform behavior, irrespective of whether the objective is EPS or EPUS.

4 Control analysis of dynamic error models

Beginning with two-step methods, we assume that a computable error estimator can be modeled near $\bar{\rho} = \mathbf{1}$ by

$$r_n = \varphi_n h_n^q \cdot \rho_{n-1}^{\delta_1}\, \rho_{n-2}^{\delta_2} = \varphi_n h_n^{q+\delta_1}\, h_{n-1}^{\delta_2-\delta_1}\, h_{n-2}^{-\delta_2}. \tag{27}$$

The validity of this structure will be demonstrated in Section 5. The objective is to construct a controller that determines ρ_n and updates the step size according to $h_{n+1} = \rho_n h_n$. The process has three parameters, (q, δ_1, δ_2), and by combining this process with a controller of the following structure,

$$\rho_n = \left(\frac{\mathrm{TOL}}{r_n}\right)^{\beta/q} \rho_{n-1}^{\alpha_1}\, \rho_{n-2}^{\alpha_2} \tag{28}$$

$$h_{n+1} = \rho_n h_n \tag{29}$$

we have three free parameters, $(\beta, \alpha_1, \alpha_2)$, to control the placement of the three poles of the closed loop. The control structure is general and follows the pattern of digital filters, [19], and we note that the elementary controller has $\beta = 1$ and $\alpha_1 = \alpha_2 = 0$. The general result is the following.

Theorem 1 *Assume that a linear multistep method has an asymptotic error estimate represented in multiplicative form as*

$$r_n = \varphi_n\, h_n^q \cdot \prod_{j=1}^{s} \rho_{n-j}^{\delta_j}. \tag{30}$$

If this method is combined with the step size control law

$$h_{n+1} = \left(\frac{\mathrm{TOL}}{r_n}\right)^{\beta/q} h_n \cdot \prod_{j=1}^{s} \rho_{n-j}^{\alpha_j}, \tag{31}$$

taking $\alpha_j = \delta_j \cdot \beta/q$ for all j, the closed loop has a single nontrivial pole at $z = 1-\beta$, with all other poles at $z = 0$. If in addition $\beta = 1$, the controller is deadbeat (finite impulse response).

Proof Taking logarithms in (30), we obtain

$$\begin{aligned}\log r_n &= (q+\delta_1)\log h_n + (\delta_2-\delta_1)\log h_{n-1} + \cdots - \delta_s \log h_{n-s} + \log\varphi_n \\ &= \left[(q+\delta_1) + \frac{\delta_2-\delta_1}{z} + \frac{\delta_3-\delta_2}{z^2} + \cdots - \frac{\delta_s}{z^s}\right]\log h_n + \log\varphi_n,\end{aligned}$$

implying that the transfer function is

$$G(z) = \frac{(q+\delta_1)z^s + (\delta_2-\delta_1)z^{s-1} + \ldots (\delta_s-\delta_{s-1})z - \delta_s}{z^s}. \tag{32}$$

In a similar fashion, for the controller we obtain

$$\left(z - (1+\alpha_1) - \frac{\alpha_2-\alpha_1}{z} - \frac{\alpha_3-\alpha_2}{z^2} - \cdots + \frac{\alpha_s}{z^s}\right)\log h_n = \frac{\beta}{q}\log\left(\frac{\mathrm{TOL}}{r_n}\right).$$

Factoring out the difference operator $z - 1$, this simplifies to

$$(z-1)(z^s - \alpha_1 z^{s-1} - \alpha_2 z^{s-2} - \cdots - \alpha_{s-1} z - \alpha_s) \log h_n = \frac{\beta z^s}{q} \log \left(\frac{\mathrm{TOL}}{r_n} \right).$$

Thus we find the control transfer function

$$C(z) = \frac{\beta z^s}{q(z-1)(z^s - \alpha_1 z^{s-1} - \alpha_2 z^{s-2} - \cdots - \alpha_{s-1} z - \alpha_s)}. \tag{33}$$

It follows that the closed loop error transfer function is

$$R_\varphi(z) = \frac{(z-1)(z^s - \alpha_1 z^{s-1} - \cdots - \alpha_s)}{(z-1)(z^s - \cdots - \alpha_s) + \frac{\beta}{q}(q+\delta_1) z^s + \cdots + \frac{\beta}{q}(\delta_s - \delta_{s-1}) z - \frac{\beta}{q}\delta_s}.$$

As for the poles, starting at $j = s$, we recursively find that the coefficient of z^{s-j} can be made zero by taking

$$\alpha_j = \frac{\beta}{q}\delta_j, \tag{34}$$

putting successively more poles at $z = 0$, until $j = 1$. If all α_j have been chosen accordingly, $z = 0$ is a pole of multiplicity s and the denominator of $R_\varphi(z)$ is reduced to

$$z^s\,(z - (1-\beta)),$$

with the last pole located at $z = 1 - \beta$. □

Remark 1 The factor $z-1$ in the numerator of $R_\varphi(z)$ represents a difference operator that removes any persistent disturbance in $\log \varphi$. This *first order adaptivity*, [19], is a consequence of the controller's integral action (29), $\log h_{n+1} = \log h_n + \log \rho_n$, which generates the step size by summing ("integrating") the step ratios.

Remark 2 Consider the elementary controller (3) in the case $s = 2$, for which $\alpha_1 = \alpha_2 = 0$ and $\beta = 1$. The poles are then determined by

$$z^3 + \frac{\delta_1}{q} z^2 + \frac{\delta_2 - \delta_1}{q} z - \frac{\delta_2}{q} = 0. \tag{35}$$

In the static error model (like in all one-step methods) $\delta_1 = \delta_2 = 0$, implying that all roots are located at $z = 0$. This makes the elementary controller *deadbeat for one-step methods*. For multistep methods, however, the elementary controller is *not deadbeat*, as $\delta_j \neq 0$. In two-step methods, if the error estimator is one order higher than the method, there is a single δ coefficient. When $\delta_1 < 0$ there are two complex conjugate poles, leading to the resonance peaks of the type already observed in Fig. 3. Although the elementary controller is stable in tandem with standard multistep methods such as AB2 and AM2, the matching is poor, inducing slowly damped oscillations in the step size sequence. As the modulus of the roots is $|z| = \sqrt{|\delta_1/q|}$, the damping is weaker in EPUS mode than in EPS.

Remark 3 The theorem implies that there is a basic class of single-parameter controllers of the form

$$h_{n+1} = \left(\frac{\mathrm{TOL}}{r_n} \cdot \prod_{j=1}^{s} \rho_{n-j}^{\delta_j} \right)^{\beta/q} \cdot h_n \tag{36}$$

that can be applied in tandem with any linear multistep method, characterized by its asymptotic error parameters q and $\{\delta_j\}_1^s$. The closed loop is stable whenever $\beta \in (0, 2)$, but in order to have a non-negative root, only $\beta \in (0, 1]$ is of interest. It is *necessary to include a step ratio compensator* to achieve deadbeat control. While a deadbeat controller is not always preferable, its existence is of significance as pole locations depend continuously on the parameters. Thus there is a neighborhood of stable controllers near $\alpha_j = \delta_j \cdot \beta/q$ and $\beta = 1$, and the full set of parameters $\{\alpha_j\}_1^s$ and β can be chosen to place the $s + 1$ poles at suitable locations.

Remark 4 Due to (36) the control problem is simplified by introducing the *dynamically compensated error*,

$$\tilde{r}_n = r_n \cdot \prod_{j=1}^{s} \rho_{n-j}^{-\delta_j}, \tag{37}$$

which *recovers the static model*, as $\tilde{r}_n \approx \varphi_n h_n^q$. In terms of the compensated error, the control system (36) above becomes

$$h_{n+1} = \left(\frac{\mathrm{TOL}}{\tilde{r}_n} \right)^{\beta/q} h_n. \tag{38}$$

This is merely a standard reduced gain integral controller, identical to the exponential forgetting used for one-step methods, [19]. By taking the gain parameter $\beta = 2/3$, which puts the nontrivial pole at $z = 1/3$, one obtains a good performance for all multistep methods operating in the asymptotic regime. This value is used in Figs. 1 and 3. Other types of controllers for one-step methods, including digital filters, [19], are also applicable to the dynamically compensated error model. As long as the step ratios are near 1, the dynamic compensator will not significantly affect the magnitude of the error, implying that $\tilde{r}_n \approx r_n$. However, the compensator will affect the dynamics and eliminate damaging resonances in the adaptive scheme.

5 Local asymptotic error estimators

Using the grid-independent representation of multistep methods in [1] the step from t_n to $t_{n+1} = t_n + h_n$ is based on constructing a polynomial $P_{n+1}(t)$ satisfying a number of structural conditions and slack conditions, such that the solution to the differential equation $y(t_{n+1})$ can be approximated by

$$y_{n+1} = P_{n+1}(t_{n+1}).$$

The error is defined by inserting the exact solution into the discretization, and satisfies

$$y_{n+1} - y(t_{n+1}) \approx c_y(\bar{\rho}_n) \cdot y^{(p_y+1)}(t_n) \cdot h_n^{p_y+1}.$$

However, this quantity is not directly computable. To obtain a reference value, we need a comparison method that produces another approximation,

$$z_{n+1} - y(t_{n+1}) \approx c_z(\bar{\rho}_n) \cdot y^{(p_z+1)}(t_n) \cdot h_n^{p_z+1}.$$

Although this quantity is not directly computable either, it follows that

$$\begin{aligned} y_{n+1} - z_{n+1} &= [y_{n+1} - y(t_{n+1})] - [z_{n+1} - y(t_{n+1})] \\ &\approx c_y(\bar{\rho}_n) \cdot y^{(p_y+1)}(t_n) \cdot h_n^{p_y+1} - c_z(\bar{\rho}_n) \cdot y^{(p_z+1)}(t_n) \cdot h_n^{p_z+1} \end{aligned}$$

is computable. Here we need to distinguish two cases.

Case 1. In the first case, the comparison method is one order higher, i.e., $p_z = p_y + 1$. For example, if the method is AB2, one can use AB3 to compute the reference value. We shall refer to this method/error estimation combination as AB2(3), where the number within paranthesis refers to the order p_z of the error estimator.

Now, if both methods operate in the asymptotic regime, the higher order result can be considered "exact," and the error estimate is effectively

$$\tilde{l}_n = \|y_{n+1} - z_{n+1}\| \approx |c_y(\mathbf{1})| \cdot \|y^{(p_y+1)}(t_n)\| \cdot h_n^{p_y+1} \cdot \hat{\pi}_l(\bar{\rho}).$$

This implies that the error constant and the dynamics $\hat{\pi}_l(\bar{\rho})$ are those from the lower order method. In effect, it emulates the situation where the actual error of the method has been computed.

Case 2. In the second case, the order of the comparison method is the same as that of the method, i.e., $p_z = p_y$. In the AB2 example, this method/error combination will be denoted AB2(2) to distinguish it from AB2(3) above. This computation can be arranged in different ways. If the solution at time t_{n+1} is represented by the value of a polynomial $y_{n+1} = P_{n+1}(t_{n+1})$, one may e.g. use the polynomial from the previous step, $P_n(t)$, and compute the extrapolated reference value $z_{n+1} = P_n(t_{n+1})$. We then have

$$\begin{aligned} y_{n+1} - z_{n+1} &= [P_{n+1}(t_{n+1}) - y(t_{n+1})] - [P_n(t_{n+1}) - y(t_{n+1})] \\ &\approx (c_y(\bar{\rho}_n) - c_z(\bar{\rho}_n)) \cdot h_n^{p_y+1} \cdot y^{(p_y+1)}(t_n), \end{aligned}$$

obtaining a raw error estimate,

$$e_n = \|y_{n+1} - z_{n+1}\| \approx |c_y(\mathbf{1}) - c_z(\mathbf{1})| \cdot \|y^{(p_y+1)}(t_n)\| \cdot h_n^{p_y+1} \cdot \hat{\pi}_e(\bar{\rho}_n),$$

where $\hat{\pi}_e(\bar{\rho})$ accounts for the dynamics of the estimator, with which the controller interacts. This function is derived in the same way as before, by obtaining a variable step size formula for the method, as well as a formula for how the previous polynomial predicts z_{n+1} through extrapolation. When the difference between these formulas are expanded in a Taylor series, we obtain e_n and $\hat{\pi}_e(\bar{\rho})$, which is approximated in the usual multiplicative way.

The raw estimates $\tilde{l}_n$ and e_n need further scaling to conform to the function w in (10). In Case 2 in EPUS mode, e_n must be scaled back to the desired estimate r_n of the method's actual error. According to (9), we want to control the magnitude of the quantity

$$\|w\| \approx \frac{|c_y(\mathbf{1})|}{b_y(\mathbf{1})} \cdot \|y^{(p_y+1)}(t_n)\| \cdot h_n^{p_y} \approx C_e \cdot \frac{\|y_{n+1} - z_{n+1}\|}{h_n \cdot \hat{\pi}_e(\bar{\rho}_n)},$$

where the estimator's error constant is given by

$$C_e = \frac{1}{b_y(\mathbf{1})} \left| \frac{c_y(\mathbf{1})}{c_y(\mathbf{1}) - c_z(\mathbf{1})} \right|. \tag{39}$$

In Case 1, the constant C_e is replaced by $C_l = 1/b_y(\mathbf{1})$ and $\hat{\pi}_e(\bar{\rho}_n)$ is replaced by $\hat{\pi}_l(\bar{\rho}_n)$. We can then describe the computational process as follows.

1. Given (t_n, y_n), compute $y_{n+1} = P_{n+1}(t_{n+1})$ at $t_{n+1} = t_n + h_n$.
2. For Case 1, take $\tilde{K} = C_l$, and for Case 2, take $\tilde{K} = C_e$, and compute the *dynamically compensated error estimate*

$$\tilde{r}_n = \tilde{K} \cdot \frac{\|y_{n+1} - z_{n+1}\|}{h_n} \cdot \prod_{j=1}^{s} \rho_{n-j}^{-\delta_j}. \tag{40}$$

3. Compute the *scaled control errors* $c_n = (\mathrm{TOL}/\tilde{r}_n)^{1/q}$, where $q = p_y$.
4. Apply recursive digital filter with coefficients $(\beta_1, \beta_2, \gamma)$ to generate ρ_n from

$$\rho_n = c_n^{\beta_1} c_{n-1}^{\beta_2} \rho_{n-1}^{-\gamma}.$$

and update the step size according to $h_{n+1} = \rho_n h_n$.

The complete system is illustrated in Fig. 4, where the classical use of the elementary controller corresponds to taking $K(z) = 1$ (no dynamic compensation) and $F(z) = 1$ (no digital filter). Note that $\hat{\pi}_e(\bar{\rho}_n)$ and $\hat{\pi}_l(\bar{\rho}_n)$ are replaced by their respective multiplicative approximants to construct the dynamically compensated error estimate. Then $\tilde{r}_n \approx \varphi_n h_n^{p_y}$ in the asymptotic regime, enabling the use of standard controllers. Without the compensator, controller performance drops, sometimes exciting resonances or instability. Table 1 gives filter coefficients for the step size controllers available in the proof-of-concept software [2].

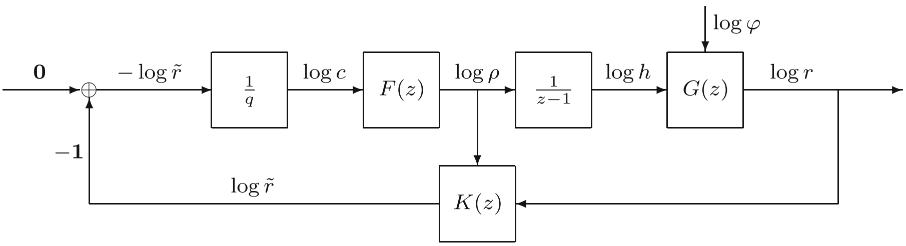

Fig. 4 A complete feedback control system with digital filter $F(z)$ and dynamic compensator $K(z)$, which transforms the error estimate $\log r$ using the step ratios $\log \rho$. The controller is decomposed into three parts. The first is a simple gain of $1/q$, converting the compensated control error into the scaled control error $\log c$. The digital filter $F(z)$ generates the step ratio $\log \rho$. Finally, a simple integrator, $1/(z-1)$, changes the step size, which enters the computational process $G(z)$. Here, we have taken $\log \mathrm{TOL} = 0$

Table 1 Filter coefficients for various controllers, [2, 21]

Filter type	β_1	β_2	γ	Property
Elementary	1	0	0	Deadbeat with compensator
Exponential forgetting	$2/3$	0	0	Medium gain I controller
PI3333	$2/3$	$-1/3$	0	Low gain PI controller
$H211$PI	$1/6$	$1/6$	0	Low-pass filter of PI type
$H211b$	$1/4$	$1/4$	$1/4$	Noise-shaping low-pass filter

For selected methods of step number $k \leq 4$, we give the parameters C_l, C_e and $\{\delta_j\}_1^s$ in Tables 2 and 3. The computation of these parameters is complex and was carried out in MATHEMATICA and MAPLE, while numerical emulations and experiments were done in MATLAB. Parameters for 5-step methods can also be provided on request. The method parameters are associated with the particular type of error estimation procedures described above, and may not work if other estimation procedures are used.

Table 2 Error coefficients and parameters for selected 2-step, 3-step and 4-step methods

Method	C_l	**Parameters** $\{\delta_j\}_1^s$		
AB2(3)	1	$-\frac{3}{5}$		
EDF2(3)	$\frac{3}{2}$	$-\frac{5}{8}$		
BDF2(3)	$\frac{3}{2}$	$-\frac{2}{3}$		
AM2(4)	1	$-\frac{2}{3}$		
dcBDF2(4)	$\frac{3}{2}$	$-\frac{2}{3}$		
AB3(4)	1	$-\frac{4}{3}$	$-\frac{10}{27}$	
EDF3(4)	$\frac{11}{6}$	$-\frac{37}{27}$	$-\frac{11}{27}$	
BDF3(4)	$\frac{11}{6}$	$-\frac{47}{33}$	$-\frac{13}{33}$	
Kregel3(4)	$\frac{44}{39}$	$-\frac{440795}{350064}$	$-\frac{135521}{388960}$	
AM3(5)	1	$-\frac{55}{38}$	$-\frac{15}{38}$	
dcBDF3(5)	$\frac{11}{6}$	$-\frac{18}{11}$	$-\frac{39}{22}$	
IDC23(5)	$\frac{3}{2}$	$-\frac{124}{85}$	$-\frac{53}{170}$	
AB4(5)	1	$-\frac{535}{251}$	$-\frac{455}{502}$	$-\frac{135}{502}$
EDF4(5)	$\frac{25}{12}$	$-\frac{7517}{3456}$	$-\frac{42377}{43200}$	$-\frac{833}{3200}$
BDF4(5)	$\frac{25}{12}$	$-\frac{67}{30}$	$-\frac{71}{75}$	$-\frac{7}{25}$
AM4(6)	1	$-\frac{43}{45}$	$-\frac{43}{45}$	$-\frac{38}{135}$
dcBDF4(6)	$\frac{25}{12}$	$-\frac{116}{45}$	$-\frac{2233}{450}$	$\frac{79}{10}$

In each method designation, the number within parenthesis refers to the order of the error estimator, which in this table is one order higher than the method order, i.e., $p_z = p_y + 1$ with $s = k - 1$. The parameters are independent of the actual choice of estimator as long as $p_z > p_y$. Method designations refer to those used in [1, 2]

Table 3 Error coefficients and parameters for selected 2-step, 3-step and 4-step methods

Method	C_e	Parameters $\{\delta_j\}_1^s$			
AB2(2)	$\frac{5}{23}$	$-\frac{45}{23}$	$-\frac{12}{23}$		
EDF2(2)	$\frac{1}{3}$	$-\frac{71}{36}$	$-\frac{5}{9}$		
BDF2(2)	$\frac{3}{5}$	$\frac{7}{30}$	0		
AM2(3)	$\frac{1}{9}$	$-\frac{26}{27}$	$-\frac{8}{27}$		
dcBDF2(3)	$\frac{3}{16}$	$-\frac{23}{24}$	$-\frac{1}{3}$		
AB3(3)	$\frac{9}{55}$	$-\frac{148}{55}$	$-\frac{182}{165}$	$-\frac{56}{165}$	
EDF3(3)	$\frac{99}{320}$	$-\frac{217}{80}$	$-\frac{183}{160}$	$-\frac{121}{320}$	
BDF3(3)	$\frac{11}{26}$	$-\frac{755}{858}$	$-\frac{212}{429}$	$-\frac{3}{22}$	
Kregel3(3)	$\frac{1683}{8986}$	$-\frac{6409445}{6326144}$	$-\frac{4579242229}{7401588480}$	$-\frac{137548701}{822398720}$	
AM3(4)	$\frac{19}{251}$	$-\frac{935}{502}$	$-\frac{415}{502}$	$-\frac{60}{251}$	
dcBDF3(4)	$\frac{99}{554}$	$-\frac{5744}{3047}$	$-\frac{7017}{6094}$	$-\frac{3159}{3047}$	
IDC23(4)	$\frac{255}{2086}$	$-\frac{1950}{1043}$	$-\frac{8837}{10430}$	$-\frac{1007}{5215}$	
AB4(4)	$\frac{251}{1901}$	$-\frac{6625}{1901}$	$-\frac{6665}{3802}$	$-\frac{3025}{3802}$	$-\frac{480}{1901}$
EDF4(4)	$\frac{45}{158}$	$-\frac{88707}{25280}$	$-\frac{567223}{316000}$	$-\frac{1647949}{1896000}$	$-\frac{155771}{632000}$
BDF4(4)	$\frac{25}{77}$	$-\frac{1229}{660}$	$-\frac{12829}{11550}$	$-\frac{3881}{7700}$	$-\frac{8}{55}$
AM4(5)	$\frac{27}{475}$	$-\frac{6548}{2375}$	$-\frac{3473}{2375}$	$-\frac{60}{251}$	$-\frac{3473}{2375}$
dcBDF4(5)	$\frac{900}{5713}$	$-\frac{79223}{28565}$	$-\frac{800936}{428475}$	$-\frac{482257}{285650}$	$\frac{152312}{28565}$

In each method designation, the number within parenthesis refers to the order of the error estimator, which in this table is the same as the method order, i.e., $p_z = p_y$ with $s = k$. The parameters are given for the specific estimation procedure obtained by using polynomial extrapolation. Method designations refer to those used in [1, 2]

6 Experimental results

Four well-known standard problems, linear as well as nonlinear, were chosen to benchmark controller performance when different two-step methods were combined with two different types of error estimators. Three-step methods were also tried, and the controllers were either the elementary controller (the reference) and three dynamically compensated controllers: exponential forgetting with gain $\beta = 2/3$, and the digital filters $H211$PI and $H211b$. All combinations were run both in EPUS and EPS modes, and for three different tolerances, for the first three problems, to check step size sequences and step ratios, as well as tolerance proportionality. In all cases the step size was selected to control the absolute error, and care was taken to ensure that the methods operated in the asymptotic regime. Initial step sizes were equally spaced, and in implicit methods the nonlinear systems were solved by Newton iteration to full accuracy.

This provided a vast evaluation material, and only a few tests can be included to illustrate the new adaptivity in practical operation. The selected examples were chosen to illustrate various performance aspects. For example, a poor combination of method and error estimator may go unstable in tandem with the elementary controller, but with a dynamic compensator the problems can be rectified. But this relies crucially on knowing the δ_j coefficients listed above for selected methods.

The first test problem is the *two-compartment dilution process*

$$\begin{aligned} y_1' &= -\frac{1}{5} y_1 \\ y_2' &= -\frac{2}{5}(y_2 - y_1) \end{aligned}$$

for $t \in [0, 20]$, with exact solution

$$\begin{aligned} y_1 &= 0.3e^{-0.2t} \\ y_2 &= 0.6(e^{-0.2t} - e^{-0.4t}). \end{aligned}$$

The initial values were chosen as the exact solution values at the first steps.

The second test problem is the *Lotka–Volterra* equation,

$$\begin{aligned} y_1' &= 0.1y_1 - 0.3y_1y_2 \\ y_2' &= 0.5(y_1 - 1)y_2 \end{aligned}$$

for $t \in [0, 62]$. The initial values were taken as $[1, 1]$, using a second order Taylor expansion around this point.

The third problem is the *van der Pol* equation,

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1 \end{aligned}$$

with $t \in [0, 20]$. The initial values were $[2, 0]$, with a second order Taylor expansion providing additional initial values. We chose $\mu = 2$ for nonstiff computation. Tolerance proportionality was checked by taking $\text{TOL} = 10^{-m}$ for a suitable integer m in each test, as well as $\text{TOL} = 10^{-m\pm q}$. Thus, when the local error per (unit) step is controlled to keep $\varphi h^q \sim \text{TOL}$, the step size should scale accordingly, like $h \sim \text{TOL}^{1/q}$, along the entire solution. All computations confirm that the control system as well as the error estimators work in accordance with theory, producing three different step size sequences one order of magnitude apart, see Figs. 5 and 6.

We finally tried stiff computation, where the method operates in part outside standard asymptotic assumptions. Thus we solved the van der Pol equation for $\mu = 10^3$ and $t \in [0, 1000]$ with the BDF2 method. Here, too, the controllers work as expected, with the best performance obtained with the BDF2/AM2 combination and the $H211$PI controller, see Figs. 7 and 8.

To also run a "large-scale" problem in moderately stiff computation, we ran a fourth problem, the 1D *Brusselator reaction–diffusion* equation,

$$\begin{aligned} u_t &= A + u^2 v - (B + 1)u + a \cdot u_{xx} \\ v_t &= -u^2 v + Bu + a \cdot v_{xx}, \end{aligned}$$

with $A = 1$, $B = 3$, $a = 0.02$, and boundary conditions $u(t, 0) = u(t, 1) = 1$ and $v(t, 0) = v(t, 1) = 3$, together with initial conditions $v(0, x) = 3$ and

$$u(0, x) = 1 + 2\,\exp(-100(x - 0.4)^2).$$

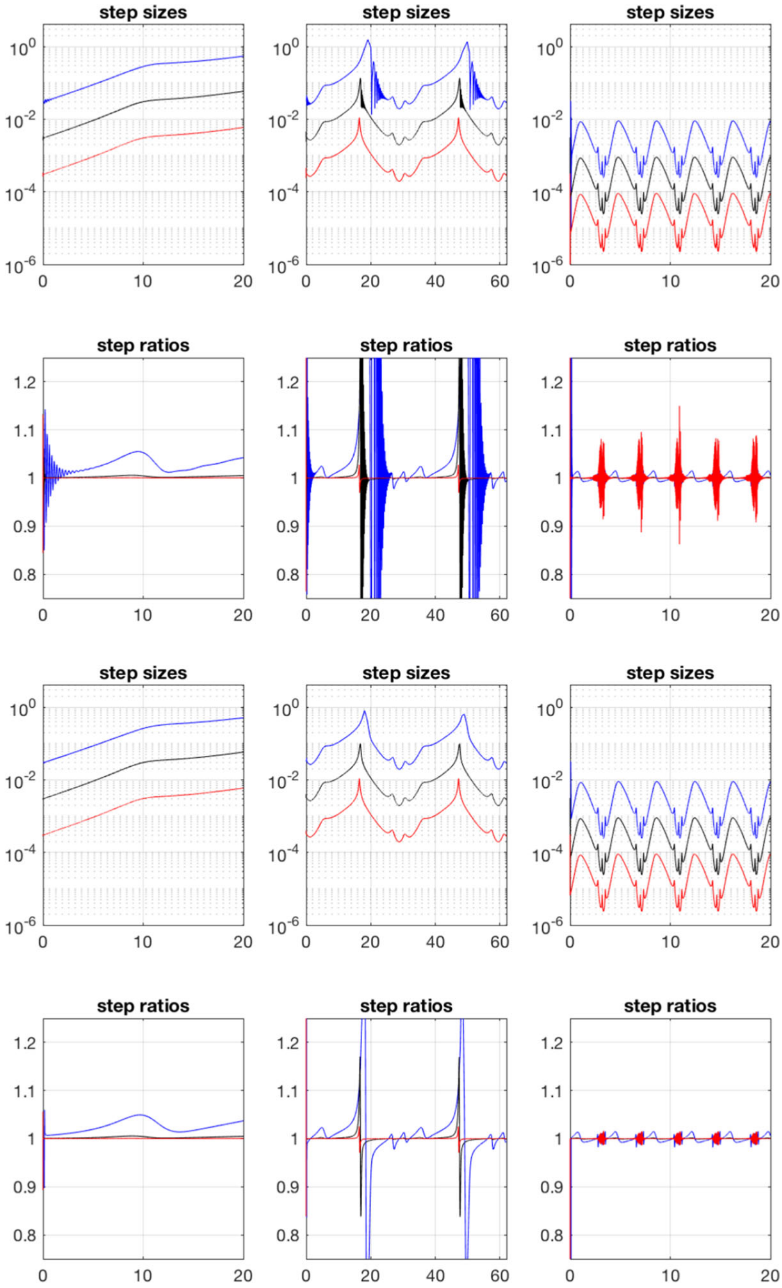

Fig. 5 The AB2(2) method, using polynomial extrapolation as error estimation, is combined with the elementary controller (upper panels) and the compensated exponential forgetting controller with gain $2/3$ (lower panels), running in EPUS mode with TOL $= 10^{-5}, 10^{-7}, 10^{-9}$. Linear problem (left), Lotka–Volterra (center), and van der Pol problem (right) all exhibit instabilities in the elementary controller in this setting, causing ringing, in particular in the Lotka–Volterra problem. The instability is overcome by the new controller, but is also less pronounced for sharp tolerances

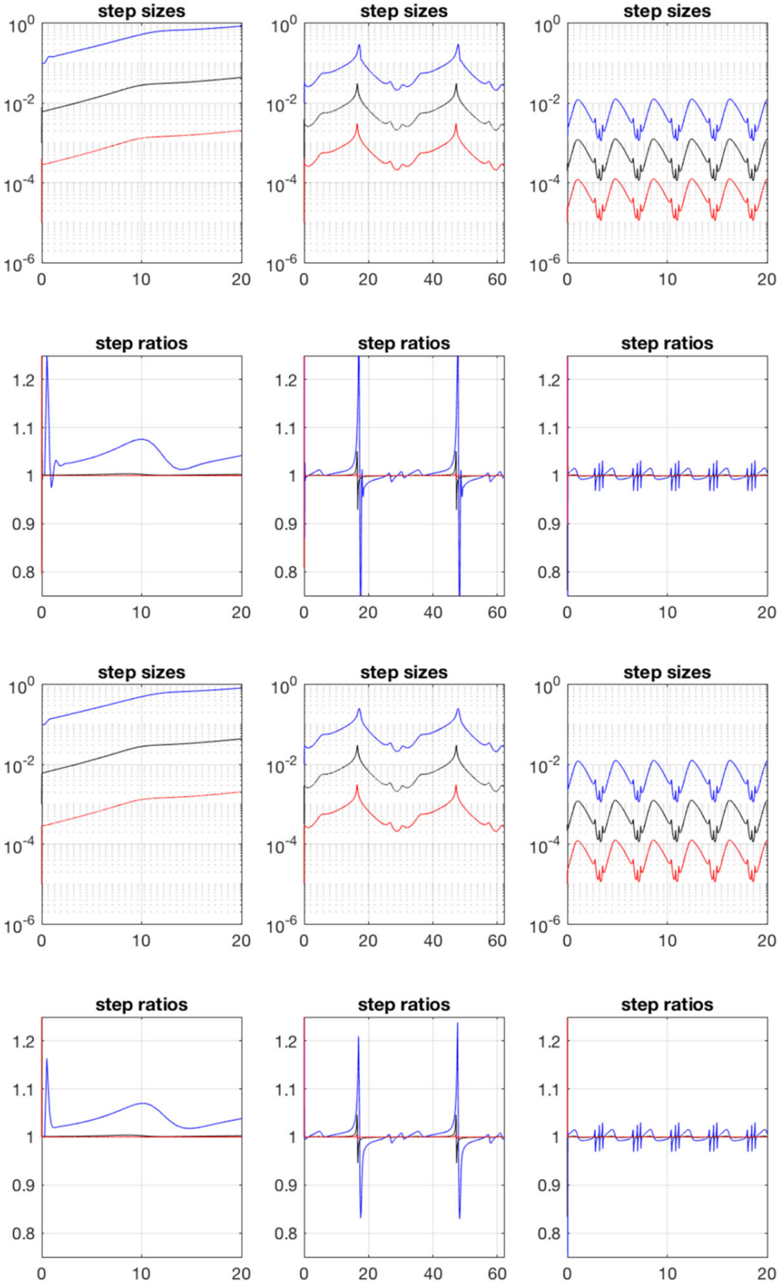

Fig. 6 The AM2(4) method, using the fourth order AB4 as error estimator, is combined with the elementary controller (upper panels) and the compensated exponential forgetting controller with gain $2/3$ (lower panels), running in EPS mode with TOL $= 10^{-4}, 10^{-8}, 10^{-12}$. Linear problem (left), Lotka–Volterra (center), and van der Pol problem (right) show good behavior with the elementary controller, but transient performance improves for the compensated controller, in particular in the linear and Lotka–Volterra problems

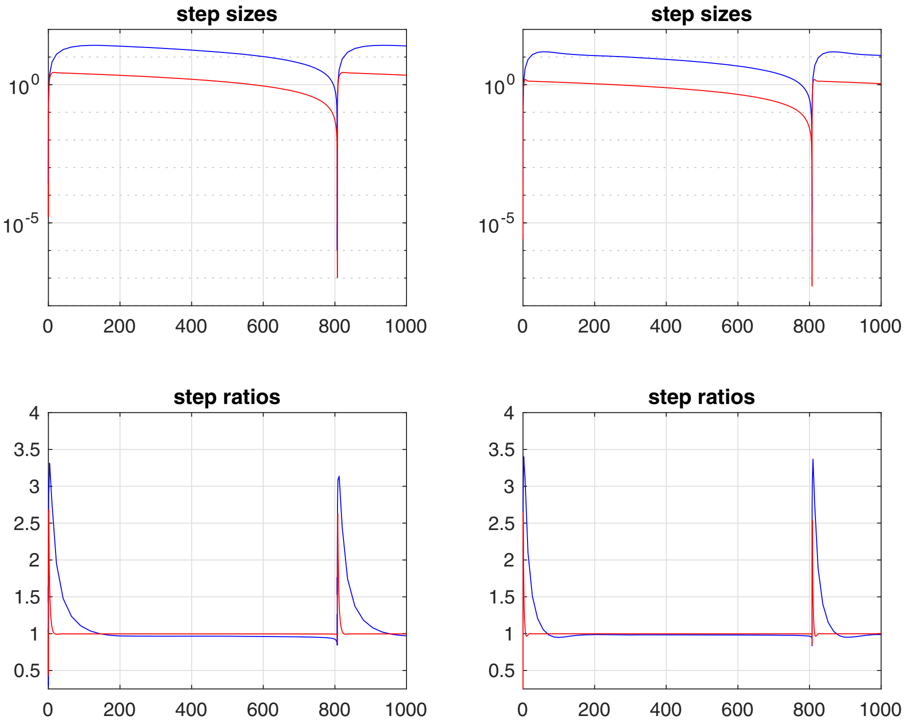

Fig. 7 Stiff van der Pol equation is solved using the BDF2 method in EPS mode and $H211$PI control with TOL $= 10^{-6}$, 10^{-9}. AM2 error estimation (left panels) is compared to 2nd order polynomial extrapolation (right panels). Step sizes vary over seven orders of magnitude and scale correctly

This problem was solved using the method of lines with a standard equidistant FDM space discretization of the diffusion term,

$$u_{xx} \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2},$$

where we chose $\Delta x = 1/(N+1)$ with $N = 60$. This setup deviates somewhat from the standard setup in the Bari test set, [15], the main differences being that we have chosen an initial condition for u that is richer in Fourier components, and we use $N = 60$ for somewhat higher resolution and increased stiffness, using a total of 120 equations. The problem was solved using the BDF2 method in EPS mode, with the error estimated by AB3, BDF2 extrapolation and AM2 methods, respectively. The absolute error was measured in the discrete L^2 norm, and the $H211PI$ controller was used at two tolerances, TOL $= 10^{-4}$ and TOL $= 10^{-7}$, to verify order and tolerance proportionality. Equidistant starting data were generated by the explicit Cash–Karp 5th order Runge–Kutta method, [4]. In this moderately stiff setup, the compensated error control still works to exacting standards, with data for six runs presented in Table 4, with the corresponding step size sequences in Figs. 9 and 10.

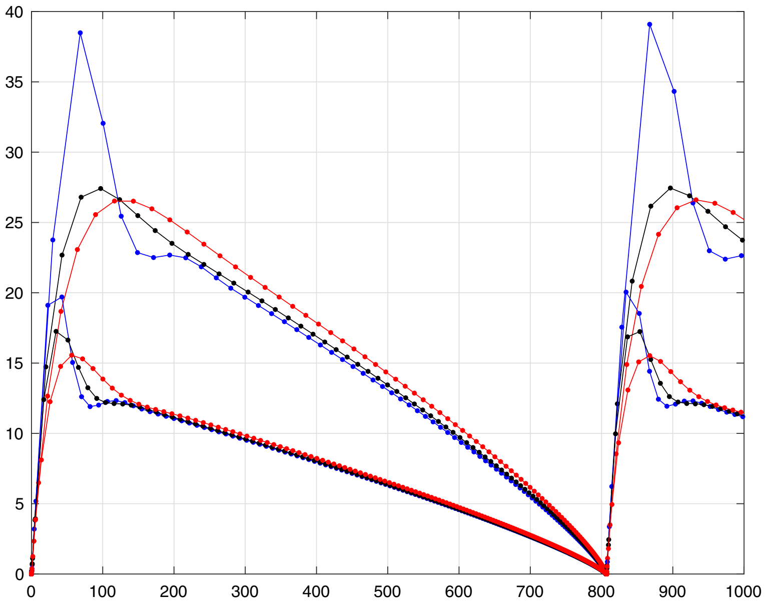

Fig. 8 Stiff van der Pol equation is solved using the same BDF2 setup as in Fig. 7, comparing different controllers. Step sizes are plotted on a linear scale for TOL $= 10^{-6}$ to reveal transient behavior and step size magnitude. Upper three curves use AM2 estimator together with the uncompensated elementary controller (blue); with compensated exponential forgetting (black); and with $H211$PI (red). Lower three curves use polynomial error estimator, and the same color coding. The AM2 estimator is twice as efficient, while the compensated $H211$PI offers the best dynamic response and smoothness

7 Conclusions

We have demonstrated that the asymptotic local error model for multistep methods has the form $r_n = \varphi_n h_n^q \cdot \hat{\pi}(\bar{\rho}_n)$, where $\bar{\rho}_n = \{\rho_{n-j}\}_{j=1}^s$ and the step ratios are defined by $\rho_{n-j} = h_{n-j+1}/h_{n-j}$, with $t_{n+1} = t_n + h_n$. The function $\hat{\pi}(\bar{\rho})$ accounts for step size history, defining the *dynamically compensated error*

$$\tilde{r}_n = r_n \cdot \prod_{j=1}^{s} \rho_{n-j}^{-\delta_j} \approx \varphi_n h_n^q,$$

where the vector $\bar{\delta} = \{\delta_j\}_{j=1}^s$ is characteristic of each linear multistep method and its error estimator. The multiplicative form of the compensator not only facilitates a linear control analysis, but is also far simpler to implement.

Table 4 Total number of steps in Brusselator problem using BDF2 in EPS mode

TOL / Method	BDF2/AM2	BDF2/BDF2	BDF2/AB3
10^{-4}	198	400	377
10^{-7}	2039	4088	3845

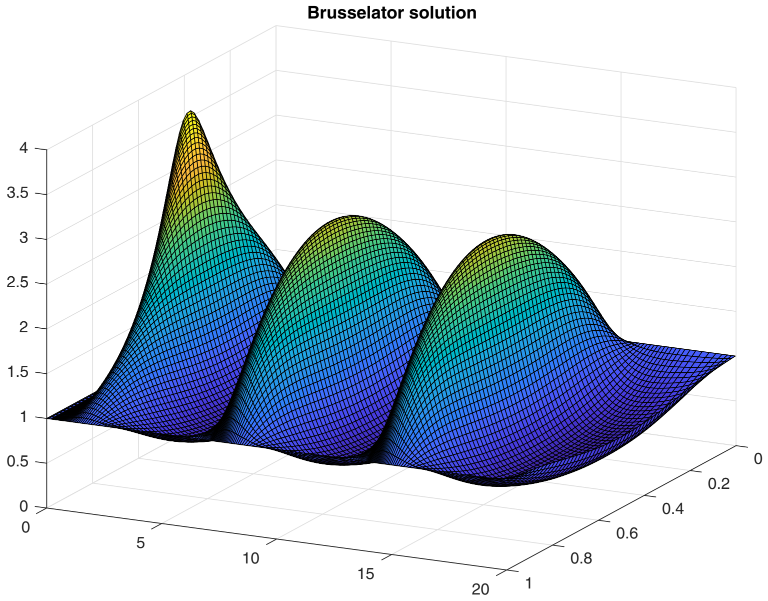

Fig. 9 Brusselator reaction–diffusion problem is solved using method of lines with adaptive BDF2 time integration for $x \in [0, 1]$ and $t \in [0, 20]$. The wave of the initial condition (left) becomes smoother with time due to diffusion

It is also shown that the conventional elementary controller may suffer damaging resonance phenomena, which are eliminated by instead controlling the compensated error. Another common issue is that many existing multistep implementations only allow the step size to be changed at fixed ratios, exciting transient behavior in the step size control. To overcome anomalies, one needs to continually adjust the step size and use a dynamic compensator.

Thus the implementation of multistep methods needs to be reconsidered. This includes specifying a method together with a dedicated error estimator and its δ_j coefficients, following a practice similar to that of Runge–Kutta methods, where an "embedded form" is used. Due to the complexity of variable step size multistep methods, we believe that this is a substantial task with a potential to improve current software standards.

Moreover, with the dynamic compensator and a well selected controller, it is possible to achieve smooth step size sequences and tolerance proportionality in EPUS mode. The implementation becomes *tolerance convergent* for Lipschitz problems. The sharper the tolerance, the smoother is the step size sequence, cf. [22]. The adaptive numerical solution converges to the exact solution as TOL → 0. As $\bar{\rho}_n \to \mathbf{1}$, the adaptive method gradually starts behaving like a constant step size method, reducing the need for the dynamic compensator.

Still, the choice of method order is not straightforward. At a given TOL, a higher order method can complete the integration in fewer steps. But fewer steps also mean a less regular step size sequence, possibly with a loss of tolerance proportionality. High

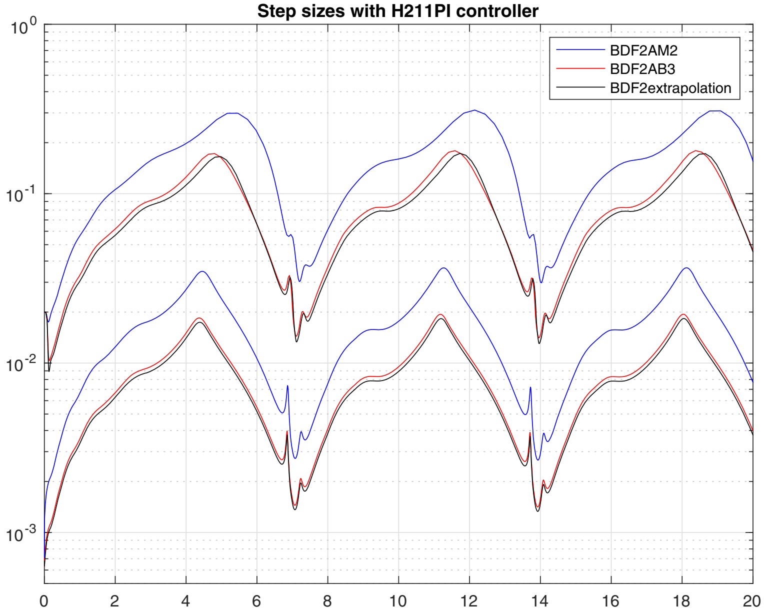

Fig. 10 Brusselator problem is solved using BDF2 in EPS mode with dynamically compensated $H211$PI control. Step sizes are plotted on a logarithmic scale for $t \in [0, 20]$. Upper three curves use TOL $= 10^{-4}$ with AM2 estimator (blue); AB3 estimator (red) and BDF2 extrapolation (black). Lower three curves use the same color coding, but with TOL $= 10^{-7}$. Notice that the first period differs from subsequent periods; compare Fig. 9. The AM2 estimator is again twice as efficient as the alternative error estimators as step sizes are twice as large. Tolerance proportionality is evident for all combinations

order methods should therefore primarily be used when the accuracy requirement is high. For a smooth step size sequence and a predictable adaptive behavior, one typically needs several hundred steps to complete the integration, meaning that the method order should primarily be chosen with respect to TOL, and less so with an aim of minimizing total number of steps. It will still offer a good work/precision trade-off.

Although the new techniques are also demonstrated to work as expected in stiff problems, further studies may be necessary for stiff computation, where the method at least in part may operate outside asymptotic conditions.

Acknowledgements Open access funding provided by Lund University. This paper was written while CA and GS were visiting the numerical analysis group at the Department of Applied Analysis of the Eötvös Loránd University in Budapest. GS was generously supported by the Hungarian Academy of Sciences, under the MTA Distuinguished Guest Fellowship Program in Hungary 2019. The organizational support from Prof. István Faragó is also gratefully acknowledged. IF was supported by EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies, funded by the Hungarian Government and the European Social Fund. Project no. ED_18-1-2019-0030 (Application-specific highly reliable ITsolutions) has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme funding scheme. The final reaction–diffusion test problem was suggested by Prof. Francesca Mazzia, who supported a visit by CA and GS to the Departments of Mathematics and Informatics, respectively, of the University of Bari, Italy, during the fall of 2019. Finally, the suggestions by two referees were very helpful in the revision of the paper and improved its significance.

References

1. Arévalo, C., Söderlind, G.: Grid-independent construction of multistep methods. J. Comput. Math. **35**, 672–692 (2017)
2. Arévalo, C., Jonsson-Glans, E., Olander, J., Selva Soto, M., Söderlind, G.: A Software Platform for Adaptive High Order Multistep Methods. ACM Trans. Math. Softw. **46**, 1–17 (2020). https://doi.org/10.1145/3372159. (Software is publicly available for research purposes)
3. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, Chichester (2008)
4. Cash, J.R., Karp, A.H.: A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides. ACM Trans. Math. Softw. **16**, 201–222 (1990)
5. Crouzeix, M., Lisbona, F.J.: The convergence of variable-stepsize, variable formula, multistep methods. SINUM **21**, 512–534 (1984)
6. Gear, C.W.: Numerical Initial Value Problems in Ordinary Differential Equations. Prentice Hall, Englewood Cliffs (1971)
7. Grigorieff, R.D.: Stability of multistep-methods on variable grids. Numer. Math. **42**, 359–377 (1983)
8. Guglielmi, N., Zennaro, M.: On the zero-stability of variable stepsize multistep methods: the spectral radius approach. Numer. Math. **88**, 445–458 (2001)
9. Gustafsson, K., Lundh, M., Söderlind, G.: A PI stepsize control for the numerical solution of ordinary differential equations. BIT Numer. Math. **28**, 270–287 (1988)
10. Gustafsson, K.: Control theoretic techniques for stepsize selection in explicit Runge–Kutta methods. ACM TOMS **17**, 533–554 (1991)
11. Gustafsson, K.: Control theoretic techniques for stepsize selection in implicit Runge–Kutta methods. ACM TOMS **20**, 496–517 (1994)
12. Hairer, E., Nørsett, S.P., Wanner, G. Solvin Ordinary Differential Equations I: Nonstiff Problems, 2nd edn. Springer, Berlin (1993)
13. Hairer, E., Wanner, G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, 2nd edn. Springer, Berlin (1996)
14. Hairer, E., Söderlind, G.: Explicit, time reversible, adaptive step size control. SIAM J. Sci. Comp. **26**, 1838–1851 (2005)
15. Mazzia, F., Magherini, C.: Test set for initial value problem solvers, release 2.4. Department of Mathematics, University of Bari, Italy. https://archimede.dm.uniba.it/~testset/testsetivpsolvers/ (2008)
16. Modin, K., Söderlind, G.: Geometric integration of Hamiltonian systems perturbed by Rayleigh damping. BIT Numer. Math. **51**, 977–1007 (2011)
17. Shampine, L., Gordon, M.: Computer Solution of Ordinary Differential Equations: The Initial Value Problem. Freeman (1975)
18. Söderlind, G.: Automatic control and adaptive time-stepping. Num. Algorithms **31**, 281–310 (2002)
19. Söderlind, G.: Digital filters in adaptive time-stepping. ACM Trans. Math. Software **29**, 1–26 (2003)
20. Söderlind, G., Wang, L.: Adaptive time-stepping and computational stability. J. Comp. Methods Sci. Eng. **185**, 225–243 (2006)
21. Söderlind, G.: Time-step selection algorithms: adaptivity, control, and signal processing. Appl. Numer. Math. **56**, 488–502 (2006)
22. Söderlind, G., Fekete, I., Faragó, I.: On the zero-stability of multistep methods on smooth nonuniform grids. BIT Numer. Math. **58**, 1125–1143 (2018)
23. Stoffer, D.: Variable steps for reversible integration methods. Computing **55**, 1–22 (1995)

Affiliations

Carmen Arévalo[1] · Gustaf Söderlind[1] · Yiannis Hadjimichael[2] · Imre Fekete[2,3]

Carmen Arévalo
cbarevalo@gmail.com

Yiannis Hadjimichael
hadjimy@cs.elte.hu

Imre Fekete
feipaat@cs.elte.hu

[1] Numerical Analysis, Center for Mathematical Sciences, Lund University, Box 118, Lund, 221 00, Sweden

[2] MTA-ELTE Numerical Analysis and Large Networks Research Group, Pázmány P. s. 1/C, Budapest, 1117, Hungary

[3] Department of Applied Analysis and Computational Mathematics, Eötvös Loránd University, Pázmány P. s. 1/C, Budapest, 1117, Hungary