Adaptivity and Computational Complexity in the Numerical Solution of ODEs

Silvana Ilie [a,2] Gustaf Söderlind [a,1]
Robert M. Corless [b,3]

[a] *Centre for Mathematical Sciences, Numerical Analysis, Lund University, Box 118 SE-221 00 Lund, Sweden*

[b] *Department of Applied Mathematics, University of Western Ontario London, Ontario, N6A 5B7, Canada*

Abstract

In this paper we analyze the problem of adaptivity for one-step numerical methods for solving ODEs, both IVPs and BVPs, with a view to generating grids of minimal computational cost for which the local error is below a prescribed tolerance (optimal grids). The grids are generated by introducing an auxiliary independent variable τ and finding a grid deformation map, $t = \Theta(\tau)$, that maps an equidistant grid $\{\tau_j\}$ to a non-equidistant grid in the original independent variable, $\{t_j\}$. An optimal deformation map Θ is determined by a variational approach. Finally, we investigate the cost of the solution procedure and compare it to the cost of using equidistant grids. We show that if the principal error function is non-constant, an adaptive method is always more efficient than a nonadaptive method.

1 Introduction

The complexity of numerical algorithms is central to the assessment of computational performance. For some algorithms, like in linear algebra, the com-

Email addresses: silvana@maths.lth.se (Silvana Ilie), Gustaf.Soderlind@na.lu.se (Gustaf Söderlind), rcorless@uwo.ca (Robert M. Corless).

[1] Supported by Swedish Research Council grant VR 621-2005-3129.
[2] Supported by a postdoctoral fellowship from the Natural Science and Engineering Research Council of Canada.
[3] Supported by a grant from the Natural Science and Engineering Research Council of Canada.

plexity is well known and established; for others, like in ordinary differential equations, the complexity is still open to analysis.

In the former case, the problems are "computable," meaning that (theoretically) the exact solution can be obtained after a finite number of operations, and this operation count then becomes a measure of the complexity. By contrast, for problems in analysis we can only compute approximate solutions converging to the exact solution. This makes an assessment of complexity more difficult, as the algorithmic complexity will depend on problem characteristics as well as the requested accuracy.

In differential equations, *adaptive algorithms* are of fundamental importance. Such algorithms attempt to minimize some (usually coarse) measure of complexity, subject to a prescribed accuracy criterion and the problem properties encountered during the computation. This is generally done by using *nonuniform discretization grids* in order to put the discretization points where they matter most to accuracy, while keeping their total number small.

Naturally, for some problems, uniform grids might be optimal from the point of view of complexity, e.g., if one considers FFT based algorithms for Poisson's equation on a rectangular domain. For linear problems, similar considerations led Werschulz to question whether adaptive methods are more efficient, using a topological argument to show that the efficiency gain would be limited to a factor of two, [24, pp. 38–39]. In this paper, however, we will prove that adaptivity is better than non-adaptivity. This result holds in a general setting for linear as well as nonlinear problems in ODEs, whenever sequential algorithms are used and the accuracy requirement is imposed in terms of some local error criterion. This is in line with computational experience, which indicates that adaptive methods are not only far superior, but often necessary in order to solve problems reasonably fast.

Because the complexity of solving an ODE numerically is a less well defined notion, and our main concern is adaptivity, we will measure the complexity as follows. Given a differential equation and a discretization method, we wish to approximate the solution on some grid such that the local error is below a prescribed tolerance TOL. What is the minimal computational cost for achieving this? In particular, how do we generate grids that achieve the minimal computational cost while maintaining the desired accuracy of the solution? Finally, in what sense is such a grid "unique," and can it be generated algorithmically?

Apart from the accuracy requirement, the grid will depend on certain problem characteristics as well as what objective is used to make the method adaptive. Some of these aspects cannot easily be dealt with. For example, if the accuracy requirement is formulated as an upper bound for the global error, then a complexity estimate will be seriously affected by the difficulty of obtaining

realistic a priori global error bounds, [16]. As this has long been understood, it is common practice that adaptive algorithms control local error estimates instead, keeping them below a preset tolerance. For this reason, we argue that a complexity analysis should start from the actual way the algorithms are constructed and implemented, instead of from the ultimate objective of global accuracy, however desirable. Furthermore, we believe that analyzing the complexity of a given problem in the proposed error criterion is a useful model for numerical practice in ODE solving. It is also a natural approach for Numerical Analysis.

This further implies that we will not deal with error accumulation in general, nor with more special cases, such as the possible cancelations of local errors in the global error accumulation. Instead, when we refer to optimal grids, it should be clearly understood that the term "optimal" is to be interpreted in the mathematical sense of optimization: a solution is *optimal with respect to a prescribed objective*. In our case the objective is based on various *local* errors. Naturally, if the objective is changed, the optimal solution changes too.

In this paper we analyze the complexity of solving ODEs using adaptive one-step methods based on local error control. The analysis is akin to the approach developed by Corless [9], but starts from a continuous representation of a local error. This accounts for the fact that when the grid points are redistributed, the local error samples will vary. Both initial value and two-point boundary value problems will be considered, and controllers generating optimal grids will be developed.

2 Grid deformations

We shall consider the problem of solving an ordinary differential equation, written as an operator equation

$$\mathcal{L}(u) = f \tag{1}$$

with either initial or boundary conditions. We seek a solution $u(t)$ on the interval $[0, T]$. For numerical computation, the original problem is typically replaced by a discrete equation

$$\mathcal{L}_{\Delta t}(y^{\Delta t}) = f \tag{2}$$

where the discretization parameter Δt represents a constant step size. The theory of such methods is well established; in particular the convergence as $\Delta t \to 0$ is of central interest, as is the order of convergence. As higher order methods often can use larger step sizes while still producing accurate results, higher order methods can be more efficient [9,13,14]. Moreover, by making

the method adaptive and using a nonuniform grid, efficiency can typically be further enhanced, by either increasing accuracy, or by reducing work, or combining both techniques.

In order to consider adaptive methods, we introduce an auxiliary independent variable τ, and a *deformation map* $\Theta : \tau \mapsto t$ such that an equidistant grid in τ produces a nonuniform grid in t. Instead of computing numerical approximations $u_k \approx u(t_k)$ for equidistant points t_k, we compute approximations $u_k \approx u\Big(\Theta(\tau_k)\Big)$ for equidistant points τ_k. Variants of such techniques are common, in particular in adaptive methods for boundary value problems, [8], [5], but also in special cases of initial value problems, see e.g. [11], [1].

In the literature, a discrete representation of a deformation map is investigated for equidistributing various monitor functions. Monitor functions may be based on local error estimates, residual estimates, global error estimates or arclength; the crucial aspect is typically that in order to obtain a convergent grid generating algorithm based on updating grid cells locally, the monitor function must only depend on the local grid cell. Such updating algorithms limit the possibilities of using global error control, which, although desirable, can often be relatively costly. For a discussion of these topics for BVP we refer the reader to [4] and references therein and for practical implementation issues and existing BVP software see, e.g., [4,5,3,7]. Recent advances on alternative monitor functions for IVP include [18] and its references. Various further issues connected with global error control may be found in, e.g., [23,15,20].

2.1 Step size modulation

Let the original independent variable $t \in [0, T]$ be written

$$t = \Theta(\tau), \tag{3}$$

with $\tau \in [0, T]$. The function $\Theta(\cdot)$ is assumed to be monotonically increasing and differentiable, and is further assumed to satisfy the boundary conditions $\Theta(0) = 0$ and $\Theta(T) = T$. Upon differentiation we have

$$\mathrm{d}t = \theta(\tau)\mathrm{d}\tau, \tag{4}$$

where we have introduced the notation $\theta(\tau) = \Theta'(\tau)$. The boundary conditions imply that the derivative θ is normalized by

$$\frac{1}{T}\int_0^T \theta(\tau)\,\mathrm{d}\tau = 1. \tag{5}$$

The discretization is now carried out using a uniform grid in τ, with the sampling correspondence $t_j = \Theta(\tau_j)$. The corresponding step sizes are (cf.

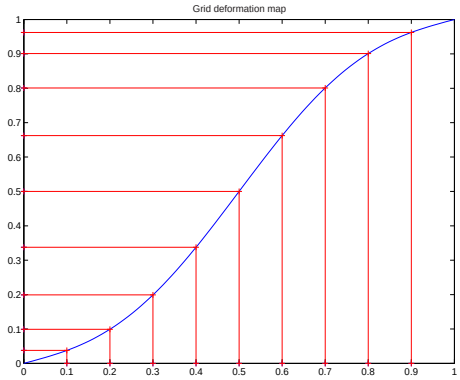

Fig. 1. *Grid deformation map.* An equidistant grid in τ (horizontal axis) is mapped to a nonuniform grid in t (vertical axis) by the function $\Theta : \tau \mapsto t$. For every choice of N there is a *unique* grid with points $t_j = \Theta(jT/N)$.

(4))

$$\Delta t_j = t_{j+1} - t_j = \Theta(\tau_{j+1}) - \Theta(\tau_j) \approx \theta(\tau_{j+1/2})\Delta\tau, \tag{6}$$

where $\tau_{j+1/2} = (\tau_{j+1} + \tau_j)/2$, with interval endpoints $\tau_0 = 0$ and $\tau_N = T$. Further, we take $\Delta\tau = \varepsilon_N$ constant, corresponding to using N steps to cover $[0, T]$. Hence $\varepsilon_N = T/N$. The derivative $\theta(\tau)$ acts as a *step size modulation function*, multiplying the discretization parameter ε_N. The grid in t is therefore nonuniform unless $\theta(\tau) \equiv 1$. We finally note that

$$T = \sum_{j=0}^{N-1} \Delta t_j = \Theta(T) - \Theta(0) = \int_0^T \theta(\tau)\,\mathrm{d}\tau \approx \frac{T}{N} \sum_{j=0}^{N-1} \theta(\tau_{j+1/2}). \tag{7}$$

In practice, an algorithm based on the technique outlined above would typically generate an approximate sequence $\vartheta = \{\vartheta_{j+1/2}\}_0^{N-1}$, with $\vartheta_{j+1/2} \approx \theta(\tau_{j+1/2})$, such that

$$\frac{1}{N} \sum_{j=0}^{N-1} \vartheta_{j+1/2} = 1. \tag{8}$$

This corresponds to the normalization requirement for the continuous step size modulation function (5). Although such an approximation is non-unique, a good approximation can usually be obtained without difficulty. Thus, in practical computation, one computes a sequence of discrete modulation functions, starting with a uniform grid, and applies standard techniques of oversampling from digital signal processing when going from one grid to the next.

Throughout the paper, we shall make extensive use of Hölder means, see e.g. [12], of both continuous and discrete functions.

Definition 1 *Let $-\infty \le s \le \infty$. The Hölder s-mean of a function $u : [0, T] \to \mathbb{R}^+$ is defined by*

$$\mathcal{M}_s(u) = \left[\frac{1}{T}\int_0^T u^s(t)\mathrm{d}t\right]^{1/s}.$$

For a positive sequence $v = \{v_j\}_0^{N-1}$, the Hölder s-mean is defined by

$$\mathcal{M}_s(v) = \left[\frac{1}{N} \sum_{j=0}^{N-1} v_j^s \right]^{1/s}.$$

Note that $\mathcal{M}_{-\infty}(u) = \min u$, and $\mathcal{M}_{\infty}(u) = \max u$. It is further worth noting that the Hölder mean equals the arithmetic mean, the geometric mean and the harmonic mean, respectively, for $p = 1, 0, -1$, and that $p = 2$ corresponds to the root mean square. We will, however, also use fractional powers p in the analysis that follows.

In terms of the Hölder means, the normalization requirements on the step size modulation function (5) and its discrete approximation (8) can be expressed as $\mathcal{M}_1[\theta] = 1$ and $\mathcal{M}_1[\vartheta] = 1$, respectively.

2.2 Grid density

It is often convenient to consider the inverse of the map Θ as a function in its own right. Let us therefore introduce the map $\Phi = \Theta^{-1}$ and note that

$$\tau = \Phi(t), \tag{9}$$

where again the function $\Phi(\cdot)$ is assumed to be monotonic and differentiable, and satisfying the boundary conditions $\Phi(0) = 0$ and $\Phi(T) = T$. Denote the derivative $\Phi'(t) = \phi(t)$. Then

$$\mathrm{d}\tau = \phi(t)\mathrm{d}t. \tag{10}$$

The boundary conditions imply that $\mathcal{M}_1(\phi) = 1$, or

$$\frac{1}{T} \int_0^T \phi(t)\, \mathrm{d}t = 1. \tag{11}$$

Comparing (10) and (4), we see that $\theta(\tau)\phi(t) \equiv 1$ (see also Figure 1) *whenever t and τ are related according to either* (3) *or* (9), i.e.,

$$\theta(\Phi(t))\phi(t) \equiv 1 \equiv \theta(\tau)\phi(\Theta(\tau)). \tag{12}$$

The function Φ is interpreted as a *grid distribution*. Using the same sampling correspondence as before, we obtain

$$\varepsilon_N = \Delta\tau = \Phi(t_{j+1}) - \Phi(t_j) = \phi(\bar{t}_{j+1/2}) \cdot \Delta t_j,$$

where $\bar{t}_{j+1/2} \in (t_j, t_{j+1})$ by the mean value theorem. Corresponding to the differential relation (10), the step size is therefore

$$\Delta t_j = \frac{\varepsilon_N}{\phi(\bar{t}_{j+1/2})},$$

where it is evident that ϕ represents a *grid density*; the step size is small when the density ϕ is large and vice versa.

Although working with the density is equivalent to working with the modulation function, there are some minor differences. In practical computation, we generate an approximate sequence $\varphi = \{\varphi_{j+1/2}\}_0^{N-1}$, with $\varphi_{j+1/2} \approx \phi(\bar{t}_{j+1/2})$, such that

$$\Delta t_j = \frac{\varepsilon_N}{\varphi_{j+1/2}}.$$

In view of (12) this sequence should be constructed so that

$$\vartheta_{j+1/2} = \frac{1}{\varphi_{j+1/2}},$$

in which case the condition $T = \sum_0^{N-1} \Delta t_j$ yields the normalization condition

$$\frac{1}{N} \sum_0^{N-1} \frac{1}{\varphi_{j+1/2}} = 1.$$

In terms of the Hölder means, the normalization of the *continuous* step size modulation function θ and grid density function ϕ are

$$\mathcal{M}_1(\theta) = 1\,; \qquad \mathcal{M}_1(\phi) = 1, \tag{13}$$

while the corresponding normalizations of their *discrete* counterparts are

$$\mathcal{M}_1(\vartheta) = 1\,; \qquad \mathcal{M}_{-1}(\varphi) = 1. \tag{14}$$

The important difference in the normalization of φ is due to the fact that while (8) is a (2nd order) numerical approximation to the integral (5), the integral (11) *cannot be directly approximated in a similar way*, due to the fact that the grid $\{t_j\}$ is non-uniform.

2.3 The nonuniform discretization

The discrete problem, on the nonuniform grid, will be denoted by

$$\mathcal{L}_{\Delta t}(y^{\Delta t}) = f.$$

We further assume, as is common, that the exact, global solution $u(t)$, sampled on the discrete grid points t_j can be inserted into the discrete problem. The sampled exact solution will be denoted by $u^{\Delta t}$. We then have

$$\mathcal{L}_{\Delta t}(u^{\Delta t}) = f - r,$$

where the residual r represents the *local truncation error*. In the sequel, we shall make the important assumptions that, *pro primo*, the step sizes suggested by the optimization procedure are permissible in the sense that they do not cause instability in the discretization scheme under consideration; *pro secundo*, the discretization parameter ε_N has been chosen so that the local truncation error behaves asymptotically as $\mathrm{O}(\varepsilon_N^p) = \mathrm{O}(N^{-p})$, where p is the order of the discretization method.

This requires that we make specific assumptions on the structure of the local truncation error. A common model is

$$r = \psi(t)\Delta t^p, \tag{15}$$

where ψ is the *principal error function*, and Δt is the *step size*. This suggests a "continuous" model of the local error, $r(t) \approx \psi(t) \cdot (\Delta t)^p(t)$, which, in terms of the grid maps discussed above, leads to the model

$$r(t) = \psi(t) \left(\frac{\varepsilon_N}{\phi(t)} \right)^p. \tag{16}$$

Here the actual choice of the *number* of grid points, N, alone determines where the functions ψ and ϕ are to be evaluated. The error can also be expressed as a function of τ by using (3), viz.,

$$r(t) = \psi(\Theta(\tau))\Big(\varepsilon_N \theta(\tau)\Big)^p. \tag{17}$$

This model applies to most discretization methods for ODEs, such as Runge–Kutta methods, as well as to finite difference methods and collocation methods for solving BVPs.

For example, the representation of the error for collocation methods in the monomial basis is, according to [2],

$$u(t) - y(t) = \Delta t_j^p \cdot u^{(p)}(t_j) P(\frac{t - t_j}{\Delta t_j}) + \mathrm{O}(\Delta t_j^{p+1}) + \mathrm{O}(\Delta t^q)$$

for $t \in [t_j, t_{j+1}]$. Here $\Delta t = \max_j \Delta t_j$. One may choose $q > p$, therefore the leading error term has a local nature and this is of the desired form. If m is the order of the ODE and ρ_ℓ with $1 \leq \ell \leq p - m$ are the canonical collocation

points, then

$$P(\xi) = \frac{1}{(p-m)!(m-1)!} \int_0^{\xi} (x-\xi)^{m-1} \prod_{\ell=1}^{p-m} (x-\rho_\ell) \mathrm{d}x \ .$$

The results of this paper apply to numerical methods for which a representation of the error is only required to satisfy (15) and so is valid for one-step methods. Further, if the error represents a local error, a local truncation error, or a residual or defect, is of no importance, as long as the error under study only depends on the local step size.

For multistep methods, the error depends also on ratios of previous step sizes to the current step size. In such a case our assumption of an error depending only on the local step size is violated, and multistep methods are not covered by the analysis.

In order to consider complexity and a corresponding grid point allocation, we need to discuss two problems:

(1) **The adaptivity problem.** For a given local error tolerance, find a grid map $\Theta(\cdot)$ (or $\Phi(\cdot)$) such that the problem can be solved to the requested accuracy, with the smallest possible number of grid points N. This is equivalent to minimizing the computational cost, subject to a prescribed accuracy requirement, by varying the grid map.
(2) **Optimal grid generation.** For a given a number of points N, find $\Theta(\cdot)$ (or $\Phi(\cdot)$) such that (some norm of) the error r is minimized.

The problems are closely related. The first is a matter of maximizing the step size without violating an accuracy requirement, while the second problem is about finding grid point locations that minimize the error. As is common, we treat the optimal grid generation problem as an optimization problem solved by a variational approach. Related work for adaptive finite element methods can be found in [6].

3 Minimization of error

We first analyze the optimal grid generation problem. This can be done either by determining $\Phi(t)$ or $\Theta(\tau)$.

3.1 *Determination of* $\Phi(t)$

For a given number of steps in a grid, N, we wish to find a function $\Phi(\cdot)$ that minimizes the local error in the $L^s[0,T]$-norm, for $1 \le s < \infty$. That is, we want to solve the optimization problem

$$\min_{\Phi} \|r\|_{L^s[0,T]} \quad \text{subject to} \quad \int_0^T \phi(t)\,\mathrm{d}t = T.$$

The norm of the continuous representation of the local error function is

$$\|r\|_{L^s} = \varepsilon_N^p \left(\int_0^T \frac{|\psi(t)|^s}{\phi(t)^{ps}} \mathrm{d}t \right)^{1/s},$$

where $|\cdot|$ is the (vector) norm at a fixed time t. The constant factor ε_N^p will not have any influence on the minimizer, and we can equivalently solve the constrained optimization problem

$$\min_{\Phi} \int_0^T \frac{|\psi(t)|^s}{\phi^{ps}}\,\mathrm{d}t \quad \text{subject to} \quad \int_0^T \phi\,\mathrm{d}t = T. \tag{18}$$

Introduce the Lagrangian $L(\Phi,\phi) = |\psi|^s/\phi^{ps} - \lambda\phi$. An optimal Φ must satisfy the Euler–Lagrange equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \phi} - \frac{\partial L}{\partial \Phi} = 0. \tag{19}$$

As the Lagrangian does not explicitly depend on Φ, this condition reduces to $\partial L/\partial\phi = \text{const.}$ where the constant is determined by the constraint. Straightforward calculation leads to the alternative characterizations

$$\hat{\phi}_s(t) = \frac{|r(t)|^s}{\mathcal{M}_1\big(\,|r(\cdot)|^s\,\big)}; \qquad \hat{\phi}_s(t) = \left(\frac{|\psi(t)|}{\mathcal{M}_{\frac{s}{ps+1}}\big(\,|\psi(\cdot)|\,\big)} \right)^{\frac{s}{ps+1}}. \tag{20}$$

Both expressions of the optimizer, denoted $\hat{\phi}_s$ for the L^s-norm, are based on the standard local error model $|r| = |\psi|\varepsilon_N^p/\phi^p$, and the second characterization also allows the limit $s \to \infty$ to be considered; for the L^∞-norm we have

$$\hat{\phi}_\infty(t) = \left(\frac{|\psi(t)|}{\mathcal{M}_{\frac{1}{p}}\big(\,|\psi(\cdot)|\,\big)} \right)^{\frac{1}{p}}. \tag{21}$$

We summarize the results obtained so far in a theorem.

Theorem 1 (L^s-optimal grid) *Let p be the order of the method, let $|\cdot|$ be a given vector norm, and let $1 \le s \le \infty$. The optimal grid generation problem*

$$\min_{\phi} \|r\|_{L^s[0,T]} \quad \text{subject to} \quad \int_0^T \phi(t)\,\mathrm{d}t = T, \tag{22}$$

where $|r(t)| = |\psi(t)|\varepsilon_N^p/\phi(t)^p$, has a unique solution

$$\hat{\phi}_s(t) = \left(\frac{|\psi(t)|}{\mathcal{M}_{\frac{s}{ps+1}}\big(\,|\psi(\cdot)|\,\big)} \right)^{\frac{s}{ps+1}}; \qquad \hat{\phi}_\infty(t) = \left(\frac{|\psi(t)|}{\mathcal{M}_{\frac{1}{p}}\big(\,|\psi(\cdot)|\,\big)} \right)^{\frac{1}{p}}. \tag{23}$$

where the optimizer $\hat{\phi}$ is independent of the number of points in the grid and of the accuracy requirement.

Further, we note that the optimal grids correspond to a grid that *equidistributes* the local error:

Theorem 2 (Equidistribution principle) *Let N be the number of grid points and let $\varepsilon_N = T/N$. For the optimal grid in the L^∞ norm, the minimum local error is $\|\hat{r}\|_{L^\infty[0,T]} = \varepsilon_N^p \hat{\nu}_\infty$, where*

$$\hat{\nu}_\infty = \mathcal{M}_{\frac{1}{p}}\big(\,|\psi(\cdot)|\,\big). \tag{24}$$

The local error is equidistributed, i.e., $|r(t)| \equiv \varepsilon_N^p \mathcal{M}_{\frac{1}{p}}\big(\,|\psi(\cdot)|\,\big)$. In the L^s norm, the minimum local error is $\|\hat{r}\|_{L^s[0,T]} = \varepsilon_N^p \hat{\nu}_s$, where

$$\hat{\nu}_s = T^{\frac{1}{s}}\ \mathcal{M}_{\frac{s}{sp+1}}\big(|\psi(\cdot)|\big).$$

The error is non-constant but satisfies the equidistribution principle

$$\Delta t_n |r(t_n)|^s \approx \varepsilon_N \mathcal{M}_1\big(\,|r(\cdot)|^s\,\big). \tag{25}$$

Proof. The assertions on the minima follow immediately by inserting the optimal $\hat{\phi}$ into the definition of the asymptotic local error model. As for the equidistribution principle, it is only necessary to consider the L^s case. Then

$$\begin{aligned}\int_{t_n}^{t_{n+1}} |r(t)|^s\,\mathrm{d}t &= \mathcal{M}_1\big(\,|r(\cdot)|^s\,\big) \int_{t_n}^{t_{n+1}} \hat{\phi}_s(t)\,\mathrm{d}t \\ &= \mathcal{M}_1\big(\,|r(\cdot)|^s\,\big) \int_{\tau_n}^{\tau_{n+1}} \mathrm{d}\tau \;=\; \mathcal{M}_1\big(\,|r(\cdot)|^s\,\big)\varepsilon_N.\end{aligned}$$

Hence we have $\Delta t_n |r(t_n)|^s \approx$ const. for the optimal grid; the local error contribution to the L^s-norm is the same on each subinterval.

3.2 Determination of $\Theta(\tau)$

The determination of Θ is analogous but leads to a different characterization. Again, we minimize the local error in the L^s norm. The optimization problem now reads, in view of the differential relation (4),

$$\min_{\Theta} \int_0^T |\psi\Big(\Theta(\tau)\Big)|^s \theta^{ps+1}(\tau)\,\mathrm{d}\tau \quad \text{subject to} \quad \int_0^T \theta(\tau)\,\mathrm{d}\tau = T, \tag{26}$$

where $\psi\Big(\Theta(\tau)\Big)$ is now considered as a function of τ. The derivation is analogous to the case for Φ, but as the Lagrangian now depends on Θ as well as θ, one would have to require that ψ is differentiable. In order to avoid this, we prefer a different approach. Note that if $ps \geq 0$,

$$\frac{1}{T}\int_0^T |\psi\Big(\Theta(\tau)\Big)|^{s/(ps+1)}\theta(\tau)\,\mathrm{d}\tau \leq \left(\frac{1}{T}\int_0^T |\psi\Big(\Theta(\tau)\Big)|^s \theta^{ps+1}(\tau)\,\mathrm{d}\tau\right)^{\frac{1}{ps+1}}. \tag{27}$$

Applying the differential relation (4), rewriting the integrals as integrals over time t, we find

$$T^{\frac{1}{s}}\mathcal{M}_{\frac{s}{ps+1}}\Big(\,|\psi(\cdot)|\,\Big) \leq \|r\|_s. \tag{28}$$

Further, equality holds in (27) if and only if $|\psi\Big(\Theta(\tau)\Big)|^{s/(ps+1)}\theta(\tau) = C$. The error is therefore minimized by $\hat{\Theta}$, characterized by the differential equation

$$\Theta'(\tau) = \left(\frac{|\psi\Big(\Theta(\tau)\Big)|^{-1}}{\mathcal{M}_{\frac{s}{ps+1}}\Big(\,|\psi\Big(\Theta(\cdot)\Big)|^{-1}\Big)}\right)^{\frac{s}{ps+1}}, \tag{29}$$

with initial condition $\Theta(0) = 0$, and where the constant is determined by the normalization requirement for Θ'. Just as in (20), the optimizer also has a representation in terms of the error,

$$\Theta'(\tau) = \frac{|r\Big(\Theta(\tau)\Big)|^{-s}}{\mathcal{M}_1\Big(\,|r\Big(\Theta(\cdot)\Big)|^{-s}\Big)}. \tag{30}$$

We summarize in the following theorems.

Theorem 3 *Let p be the order of the method, let $|\cdot|$ be a given vector norm, and let $1 \leq s < \infty$. The optimal grid generation problem*

$$\min_{\Theta} \|r\|_{L^s[0,T]} \quad \text{subject to} \quad \int_0^T \Theta'(\tau)\mathrm{d}\tau = T, \tag{31}$$

where $|r(t)| = |\psi(\Theta(\tau))|\varepsilon_N^p \Theta'(\tau)^p$, has a unique solution, $\hat{\Theta}_s$, satisfying (29).

The corresponding result for the L^∞ norm is

$$\hat{\Theta}'_\infty(\tau) = \left(\frac{|\psi\big(\hat{\Theta}_\infty(\tau)\big)|}{\mathcal{M}_{\frac{1}{p}}\big(|\psi(\cdot)| \big)} \right)^{-\frac{1}{p}} \tag{32}$$

with initial condition $\Theta_\infty(0) = 0$. The optimizer is independent of the number of points in the grid and of the accuracy requirement.

As before, the optimal grids correspond to a grid that equidistributes the local error. The minima $\hat{\nu}_\infty$ and $\hat{\nu}_s$ remain identical to those given in Theorem 2. The equidistribution is different in the L^s norm, however.

Theorem 4 (Equidistribution principle) *Let N be the number of grid points and let $\varepsilon_N = T/N$. For the optimal grid in the L^s norm, the error is non-constant but satisfies the equidistribution principle*

$$\Delta\hat{\Theta}_n |r(\hat{\Theta}(\tau_n))|^s \approx \varepsilon_N \mathcal{M}_1\Big(|r(\hat{\Theta}(\cdot))|^{-s} \Big). \tag{33}$$

Let $\Delta\tau$ be the uniform auxiliary step size. The step size on the nonuniform grid which minimizes the error is generated with the optimal step modulation function, $\hat{\theta}$, delivered by Theorem 3,

$$\Delta t(t) = \hat{\theta}(\tau)\Delta\tau \ .$$

To this step size there corresponds a local error

$$r(t) = |\psi(t)|\Delta t^p(t) = \mathcal{M}_{\frac{1}{p}}\Big(|\psi(\cdot)| \Big)\Delta\tau^p \tag{34}$$

which is constant along the interval of integration.

A *discrete* formulation of the minimax version of Theorem 3 has been given in [9,10]. The reason for preferring the *continuous* representation (16) here, is that the optimization problem then has a unique solution which is always *independent of the number of sampling points as well as of their actual locations*. In particular, in the continuous setting the Hölder mean of the local principal error function, $\mathcal{M}_{\frac{1}{p}}(|\psi(\cdot)|)$, is independent of the grid, being a characteristic of the problem, the method and the order only. The solution can therefore be used for any accuracy requirement and generate grids with any desired number of points.

Finally, we note that the constraint together with (32) leads to

$$\mathcal{M}_{\frac{1}{p}}\Big(|\psi(\cdot)| \Big) = \mathcal{M}_{-\frac{1}{p}}\Big(|\psi(\hat{\Theta}(\cdot))| \Big).$$

This is in full agreement with the discrete formulation in [9].

3.3 Numerical computation of Θ (Initial Value Problems)

Comparing the characterizations of $\hat{\Phi}_\infty$ and $\hat{\Theta}_\infty$, we note that the former can in effect be approximated directly, while the latter requires the solution of the differential equation (32). We therefore believe that Φ is better suited for adaptive BVP solvers, with Θ being preferred for IVPs.

Let us consider the numerical solution of (32), rewritten as

$$\Theta' = \gamma \cdot |\psi(\Theta)|^{-1/p}, \tag{35}$$

where γ is constant. Using the explicit Euler method, noting that the independent variable is τ and using step size $\Delta\tau = \varepsilon_N$, we obtain the recursion

$$\Theta_{n+1} - \Theta_n = \varepsilon_N \cdot \gamma \cdot |\psi(\Theta_{n-1})|^{-1/p}.$$

As $\Theta_n = t_n$ and $\varepsilon_N \cdot \gamma = \text{TOL}^{1/p}$, we have

$$\Delta t_n = \left(\frac{\text{TOL}}{|\psi_{n-1}|} \right)^{1/p} , \tag{36}$$

with $\psi_{n-1} \approx \psi(t_{n-1})$. Since $\psi_{n-1}^{-1/p} = \Delta t_{n-1} r_{n-1}^{-1/p}$ we derive

$$\Delta t_n = \left(\frac{\text{TOL}}{|r_{n-1}|} \right)^{1/p} \Delta t_{n-1} , \tag{37}$$

which we recognize as the elementary deadbeat controller; this is the most frequently used step size controller in IVP solvers. Under additional smoothness conditions on ψ, other, more advanced controllers can be seen to be compatible with the solution of the differential equation defining Θ. These include PI, PID controllers and digital filters, see [21].

3.4 Numerical computation of ϕ (Boundary Value Problems)

Let us briefly describe a computational process for generating the grid in a BVP, [19]. As no a priori information is available, one starts from an equidistant grid. This is usually coarse and has a relatively small number of grid points, say $N = 30$. A new, nonuniform grid is generated by solving the BVP on the coarse, uniform grid, and obtaining an error estimate on that grid. The error estimate is then used to update the grid and construct a nonuniform grid. Ideally, only one such update should be carried out, but the subsequent error estimate obtained on the nonuniform grid will indicate whether more updates are needed.

Let φ denote the original, uniform grid. After solving the problem on this grid, we obtain an error estimate r on this grid. This vector is non-constant. Under the assumption $|r_j| = |\psi_j|\Delta t_j^p$, together with $\Delta t_j = \varepsilon_N/\varphi_{j+1/2}$, the simplest update of φ is

$$\hat{\varphi} := \varphi \cdot |r|^{1/p}\gamma, \tag{38}$$

where the dot indicates pointwise multiplication of the vectors, and where the scalar γ is chosen so that the normalization $\mathcal{M}_{-1}(\hat{\varphi}) = 1$ is preserved. Note that this update can only be successful provided that the error depends exclusively on the local step, and that the update is the BVP counterpart to the deadbeat controller (37) for IVPs. More advanced controllers, such as those in [21] may also be employed.

Finally, and most importantly, we note that if the updating (38) is applied repeatedly, the density φ keeps changing *until the error* $|r|$ *is constant.* In other words, when this grid generating scheme is convergent, it converges to an equidistributed error. Naturally, the control algorithm must check for this convergence.

Once the new density function has been determined, we also determine the necessary number of steps, $\hat{N}$, such that $\|r\|_\infty = \text{TOL}$. It then remains to generate the grid with $\hat{N}$ points; this is accomplished by *oversampling* the sequence φ from N to $\hat{N}$ points. Here we note that, as the (continuous) deformation and density functions are *independent* of the accuracy requirement, the "shape" of φ remains unchanged when changing from N to $\hat{N}$ points.

A good way of oversampling φ is to use standard techniques from signal processing, e.g. spline interpolation. The spline is evaluated at $\hat{N}$ equidistant points. Note that this step makes use of the fact that the sequence φ can be mapped from the independent t variable, to the independent τ variable *without affecting its function values.* (This means that we identify φ with $1/\vartheta$, see (12), and actually perform the oversampling on ϑ, [19].) Therefore, equidistant interpolation is sufficient, and once the prolonged vector φ is found, the grid points as well as step sizes are uniquely determined by this process, although a different choice of oversampler may produce a slightly different grid.

It is clear that there could be several grids that produce solutions within a given tolerance, and it is therefore important to clarify in what sense there is a "unique" optimal grid. The following facts of the adaptivity studied here are of importance:

- Given an ODE with a sufficiently smooth exact solution, and a discretization method, there is a smooth, continuous principal error function (monitor function), associated with that method and the exact solution of the ODE
- A smooth monitor function gives rise to a unique, monotone, C^0 modulation function $\theta(\tau)$, or density function $\phi(t)$, which is *optimal with respect to local*

error control in the sense that it satisfies the Euler–Lagrange equations

- Given a continuous modulation function $\theta(\tau)$, there is, for every N, a *unique* grid obtained by equidistant sampling of θ in the τ variable
- The continuous modulation function θ is *independent of* N, i.e., the same modulation function is optimal for all accuracy requirements
- A grid generation process will *approximate* the N samples of the continuous modulation function θ, at equidistant points in τ, by a discrete sequence $\{\vartheta\}$
- To every discrete approximation $\{\vartheta\}$ there corresponds a *unique* grid, with unique, computable step sizes
- Although there are many approximating sequences, a *convergent* process ($N \to \infty$) produces *similar grids*, in the very same sense as a convergent discretization method produces similar numerical solutions to an ODE for different step sizes

Thus, although no discrete approximation to a continuous function is ever unique, an approximation process converging to a unique limiting function will, for N sufficiently large (as needed to accurately reproduce the solution of the differential equation), produce similar grids that effectively lead to the same computational results. For "optimal grids," the only issue of practical significance is therefore that the limiting, continuous modulation (density) function is unique.

4 Minimization of computational cost

In practical computations, it is common to prescribe an upper bound for the local error, TOL, and construct controllers that *equidistribute* the local error. The adaptivity problem is to find Θ (or Φ) such that the problem is solved, to the requested accuracy, *with the minimum number of grid points*. For local error control, this problem has the same solution as the optimal grid generation problem, see Theorem 2.

4.1 Minimizing the number of grid points

Take a grid of N points, with a corresponding $\varepsilon_N = T/N$. If the solution is smooth enough and N sufficiently large to produce an asymptotic behavior, then the norm of the local error on the non-uniform grid can be expressed as $\|r\|_s = \varepsilon_N^p \nu_s$. If we require that $\|r\|_s \leq \text{TOL}$, then the necessary number of steps in the grid is

$$N_s \approx T \left(\frac{\nu_s}{\text{TOL}} \right)^{1/p}. \tag{39}$$

As T and TOL are constants, the minimum number of grid points $\hat{N}$, is obtained for the minimum value of ν_s, denoted by $\hat{\nu}_s$. From Theorem 2, we obtain the following result.

Theorem 5 (Minimum number of grid points) *The minimum number of steps in a grid to solve the problem (1) to accuracy* TOL *is*

$$\hat{N}_s = T^{1+\frac{1}{sp}} \left(\frac{\mathcal{M}_{\frac{s}{sp+1}}(|\psi(\cdot)|)}{\text{TOL}} \right)^{1/p}; \qquad \hat{N}_\infty = T \left(\frac{\mathcal{M}_{\frac{1}{p}}(|\psi(\cdot)|)}{\text{TOL}} \right)^{1/p}. \tag{40}$$

4.2 Time complexity

In the model of computation considered in this paper, we ignore memory hierarchy, overheads and interpolation costs, as it is common in the standard theory of information-based complexity, see e.g. [24]. The time complexity of the algorithm will be measured either by the number of function evaluations or by the number of arithmetic operations. Furthermore, we assume a sequential model of computation. We consider the following cost model $W = c(p)N$, where N is the number of grid points and the cost per step, $c(p)$, is a constant depending on the method, its order and the dimension of the problem. This model applies to both IVPs and BVPs.

Assuming that no steps are rejected, the cost of an algorithm of constant order for initial value problems is linear in the size of the grid, i.e. $\mathrm{O}(N)$. On the other hand, methods for BVPs require solving some linear system of dimension kN. The $kN \times kN$ matrix of the system is typically a "band matrix" (e.g., almost block-diagonal matrix for collocation methods) with a fixed bandwidth, depending only on the order of the method and the dimension of the differential system. Therefore the cost of these methods is also $\mathrm{O}(N)$.

By direct application of Theorem 5 the minimum cost to solve the problem (1) with the given method at order p while the local error satisfies the accuracy TOL in the L^s–norm, is

$$\hat{W}_s = c(p) T^{1+\frac{1}{sp}} \left(\frac{\mathcal{M}_{\frac{s}{sp+1}}(|\psi|)}{\text{TOL}} \right)^{1/p}, \tag{41}$$

while in the L^∞–norm, the minimum cost is

$$\hat{W}_\infty = c(p)\, T \left(\frac{\mathcal{M}_{\frac{1}{p}}(|\psi|)}{\text{TOL}} \right)^{1/p}. \tag{42}$$

This may be used to estimate how much more efficient an adaptive method is

compared to a nonadaptive method, solving the problem to the same accuracy on a uniform grid. Using a fixed step size, the necessary number of grid points is, for the L^∞ norm, $N_{fix,\infty} = T\||\psi|\|_\infty^{1/p}\,\mathrm{TOL}^{-1/p}$. Therefore the cost of the algorithm on a uniform grid is

$$W_{fix,\infty} = c(p)\,T\left(\frac{\||\psi|\|_\infty}{\mathrm{TOL}}\right)^{1/p}.$$

The efficiency of adaptivity can now be calculated as the ratio between the cost associated with the adaptive method and that of the fixed-grid method.

Theorem 6 (Efficiency gain) *The efficiency gain due to adaptivity is*

$$\frac{W_{fix,\infty}}{\hat{W}_\infty} = \left(\frac{\||\psi|\|_\infty}{\mathcal{M}_{\frac{1}{p}}(|\psi|)}\right)^{1/p}. \tag{43}$$

The gain depends only on the method order p and on the principal error function $|\psi|$. It is independent of the accuracy requirement and the number of grid points.

Remark. Note that for $p > 0$ it holds that

$$\mathcal{M}_{\frac{1}{p}}(|\psi|) \le \mathcal{M}_\infty(|\psi|) = \||\psi|\|_\infty, \tag{44}$$

with equality if and only if $|\psi|$ is a constant function. (In that exceptional case, local error equidistribution occurs on a uniform grid.) From Theorem 6 above it therefore follows that, if $|\psi|$ is nonconstant, *an adaptive method is always more efficient than a nonadaptive method.* This supports "conventional wisdom" and resolves the complexity controversy, [24, p. 124]. Moreover, the efficiency gain may be arbitrarily large, as will be demonstrated in the next section.

The cost of finding the optimal grid is not covered by the analysis above. For algorithms solving IVPs, local error equidistribution is very inexpensive, and existing step size controllers, see e.g. [21,22], closely approximate the optimal grid. However, there are special classes of problems, where the usual step size selection schemes interfere with or destroy desirable properties, e.g. in geometric integrators for time-reversible problems. Reversible step size controllers are then successfully used to overcome this difficulty [11], but may not be optimal in the sense of the present analysis. Nevertheless, adaptivity is known to pay off also in such cases.

For solving BVPs with global methods, current codes generate a sequence of grid and solution computations before choosing a fine enough grid on which the solution meets the desired accuracy, see Section 3.4. The convergence of

the sequence of grids is monitored, and is normally observed in practice for carefully selected monitor functions, [19]. Related theoretical results concerning convergence of adaptive grids for finite element methods can be found in [17].

For efficiency, in practice most of the Newton iterations are done on coarse grids such that only one Newton iteration is needed on the final grid. Suppose n such coarse grids are generated and on each grid i, of size N_i, s_i Newton iterations are performed. The relative cost of generating the equidistributed grid, defined as the ratio between the cost of equidistribution to the cost of computing the solution on the optimal grid, is therefore $(1/\hat{N}_\infty)\sum_{i=1}^{n} s_i N_i$. A fraction of the efficiency gain may be used to cover the relative cost of equidistribution, thus adaption again pays off. We note that the efficiency gain may be arbitrarily large (depending on the problem).

5 Numerical results

We illustrate the results by demonstrating their implications for a few simple computational problems.

Example 1. We first consider the theory's implications for a scalar initial value problem,

$$y' = \lambda(y - A\mathrm{e}^{\mathrm{i}\omega t}) + \mathrm{i}\omega\, A\mathrm{e}^{\mathrm{i}\omega t}; \quad y(0) = 0, \tag{45}$$

with solution $y(t) = A(\mathrm{e}^{\mathrm{i}\omega t} - \mathrm{e}^{\lambda t})$, for some amplitude A and a real $\lambda < 0$. In order to separate the treatment of the transient and the steady state solution, we first consider the case $\omega = 0$, for which $y = A(1 - \mathrm{e}^{\lambda t})$, and $y^{(p)} = -\lambda^p A\mathrm{e}^{\lambda t}$. We assume that the local error is of the form $\psi = \kappa_p y^{(p)}$, which here gives $\psi = -\kappa_p \lambda^p A\mathrm{e}^{\lambda t}$, where κ_p is the "error constant" of the method. Theorem 5 then gives

$$\hat{N}_\infty = p\left(\frac{\kappa_p A}{\mathrm{TOL}}\right)^{1/p} \cdot (1 - \mathrm{e}^{\lambda T/p}). \tag{46}$$

When $|\lambda T|$ is small, this can be approximated by

$$\hat{N}_\infty \approx |\lambda T|\left(\frac{\kappa_p A}{\mathrm{TOL}}\right)^{1/p}, \tag{47}$$

implying that work grows approximately linearly in time during the non-stiff phase. This also represents the work when a constant step size is used to integrate the problem T units of time.

For a stiff problem, however, when $\lambda T \ll -1$, and the transient decays into

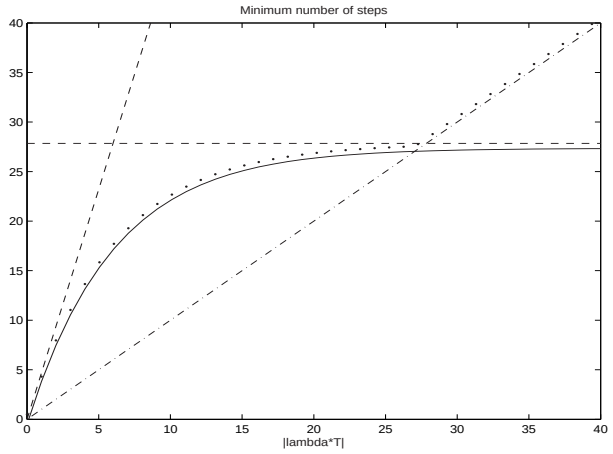

Fig. 2. *Minimum number of steps in model problem (45).* Steep dashed line on the left indicates the necessary number of steps vs $|\lambda T|$ for a nonadaptive method to keep the local error less than TOL. This equals the number of steps for short range integration, (47). Solid curve indicates required number of steps for a stiff adaptive method, (46). The upper bound is indicated by the horizontal dashed asymptote. Dotted curve indicates the number of points needed by an adaptive non-stiff method, which for stability reasons must use step sizes $|\Delta t\lambda| \leq \mathrm{O}(1)$. This stability condition is indicated by the dash-dotted slanted line to the right. Both the non-stiff and the stiff adaptive methods are seen to be more efficient than a constant-step method. However, if $|\lambda T|$ is large, the stiff adaptive method becomes vastly more efficient than any one of the alternatives.

an equilibrium, we find that

$$\hat{N}_\infty \leq p \left(\frac{\kappa_p A}{\mathrm{TOL}} \right)^{1/p}, \tag{48}$$

as long as the requested step sizes do not lead to numerical instability. This implies that *a finite number of steps is sufficient* for an adaptive stiff method, *no matter how long the range of integration is.* For $\lambda T/p \ll -1$, the efficiency gain, given by the ratio of (47) to (48), is therefore approximately $|\lambda T/p|$ and can be arbitrarily large. The bounds are illustrated in Figure 2 for $\kappa_p A = 1$, $\mathrm{TOL} = 10^{-4}$ and $p = 6$.

For the steady state solution $y = A\mathrm{e}^{\mathrm{i}\omega t}$ the situation is different. Take $y(0) = A$ to eliminate the transient. A similar calculation then shows that

$$\hat{N}_\infty = |\omega T| \left(\frac{\kappa_p A}{\mathrm{TOL}} \right)^{1/p}. \tag{49}$$

Qualitatively similar to (47), this is independent of the magnitude of $|\lambda T|$ as long as the requested step sizes do not lead to numerical instability, but the number of steps grows linearly with time T. If $|\omega/\lambda| \ll 1$, the problem is stiff and the number of points behaves like the dotted curve in Figure 2, although the time step will be limited by ω and not by λ as in the case of a non-stiff

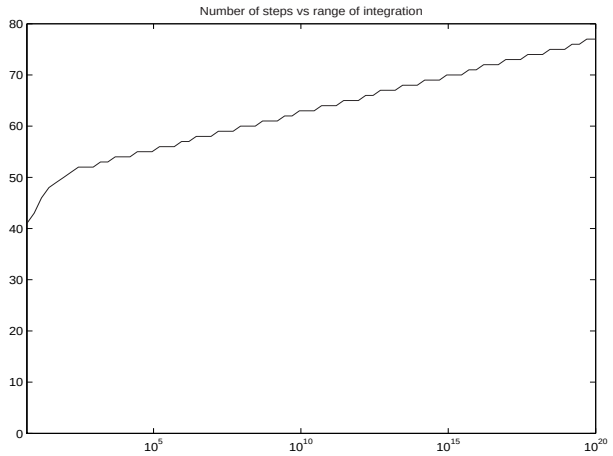

Fig. 3. *Number of steps as a function of the range of integration.* When an adaptive stiff solver is used to solve a chemical kinetics problem that settles into equilibrium, work is plotted as a function of integration end point T. The number of steps for $T = 5$ is 42, while 78 points are used for $T = 10^{20}$.

solver. The efficiency gain due to adaptivity is then essentially determined by the ratio $|\lambda/\omega|$, as long as $\lambda T \ll -1$.

Example 2. Because Example 1 above is only a theoretical model problem, we solved a chemical kinetics problem,

$$y_1' = 1 - y_1 - \frac{m y_1 y_2}{a + y_1}$$
$$y_2' = \frac{m y_1 y_2}{a + y_1} - y_2$$

with $m = 0.16$ and $a = 0.25$, and initial conditions $y_1(0) = 0.5$ and $y_2(0) = 0.02$, respectively, on the interval $[0, T]$. The solution settles to an equilibrium after some 5 time units. The problem was solved using MATLAB's stiff solver `ode23s`, with default settings, over time intervals ranging from $T = 5$ to $T = 10^{20}$, in order to demonstrate the qualitative behavior predicted by (48) in an adaptive solver. The number of steps used by the code is shown as a function of the choice of endpoint T in Figure 3. According to theory, the work needed to meet the tolerance is bounded. In practice, the code uses some safety measures to restrain the step size and make sure that a solution can be plotted. But this has only minor effects; the necessary number of steps for the interval $[0, 5]$ is 42, and less than twice the number of steps are needed for the interval $[0, 10^{20}]$.

Example 3. In the previous example total work for integrating a transient is finite, independent of the range of integration. This result is however only practically relevant in a problem that does not settle into an equilibrium. Consider the van der Pol equation with initial values $y_1(0) = 2$ and $y_2(0) = 0$,

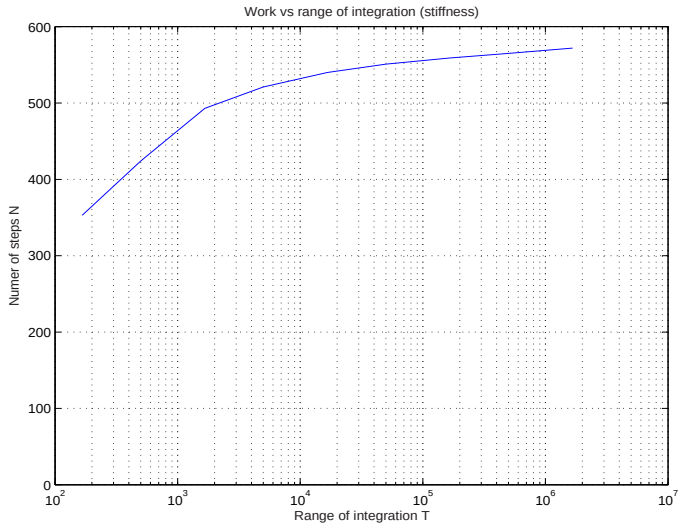

Fig. 4. *Stiff van der Pol problem.* The number of steps N needed to cover a full period $[0, T]$ is plotted vs integration range $T = 5\mu/3$, where the stiffness controlling parameter $\mu \in [10^2, 10^6]$. Total work is effectively bounded independent of of μ as predicted by theory.

$$y_1' = y_2$$
$$y_2' = \mu \cdot (1 - y_1^2)y_2 - y_1$$

for values of μ in the range $[10^2, 10^6]$. The problem has a periodic limit cycle with period just short of 2μ, and the problem becomes stiffer for larger values of μ. A step size $\Delta t \sim \mathrm{O}(1/\mu)$ is needed in order to resolve the sharp transition regions; and only an adaptive method can ever exceed such a step size. A good stiff solver can, on the other hand, reach step sizes as large as $\mathrm{O}(\mu)$ during the phases when the solution is in a quasi-equilibrium. This indicates that while a nonadaptive method needs on the order of $\mathrm{O}(\mu^2)$ steps to solve the problem, an adaptive method should be able to solve the problem in a finite number of steps, independent of μ, if our claims are correct. (This also implies that the efficiency gain for adaptation is a most significant factor on the order of $\mathrm{O}(\mu^2)$). Figure 4 shows the number of steps used by `ode23s`, run with default settings, plotted vs. the range of integration when a full period of the solution was computed.

Example 4. Finally, a simple adaptive two-point BVP solver was implemented in MATLAB for solving the problem

$$u' = v$$
$$v' = A \cdot \sin(\omega t)$$

subject to the boundary conditions $u(0) = u(1) = 0$, and with the parameters $A = 10$, $\omega = 10$, at a local error tolerance $\mathrm{TOL} = 10^{-5}$. This is a simple

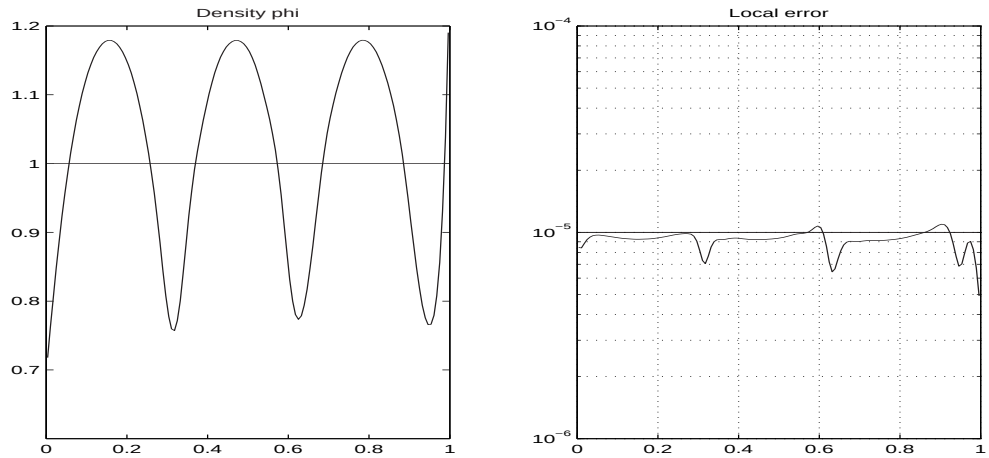

Fig. 5. *Adaptive two-point BVP solver.* Left graph shows initial uniform grid ($\phi(t) \equiv 1$) as well as how the optimized grid density $\hat{\phi}_\infty(t)$ varies on $[0, 1]$. error on the nonuniform grid, nearly equidistributed on $[0, 1]$. Further grid updates bring the local error closer to equidistibution, but the added benefits are marginal.

quadrature problem and may be considered as a 1D "Poisson equation." It was solved on a coarse, uniform grid of 30 points, using the midpoint method. The coarse grid is employed to obtain a local error estimate, which is then used to calculate the optimal $\hat{\varphi}_\infty$. The error magnitude determines the necessary number of grid points, $\hat{N}_\infty$, for the tolerance TOL. The nonuniform grid is constructed by oversampling $\hat{\varphi}_\infty$ from the original 30 points to $\hat{N}_\infty = 136$ points, as determined by the accuracy requirement. Finally the problem is solved on that nonuniform grid. Figure 5 shows the results. As 161 steps would have been necessary on a uniform grid, the adaptive method yields an efficiency gain of 18%.

6 Concluding remarks

In this paper we have studied the computational cost of adaptive methods for ODEs. In particular we study adaptive techniques based on various local error estimates to control the step size, in order to give an analysis that reflects computational practice.

Contrary to previous claims, we show that such adaptive techniques are always beneficial, and that the efficiency gain is given by the ratio $\||\psi|^{1/p}\|_\infty / \mathcal{M}_1(|\psi|^{1/p})$ which is always greater than one, and is potentially arbitrarily large.

However, it is only problems with widely varying time constants (stiff problems) or with steep gradients (e.g. BVPs with steep boundary layers) that will be solved in a vastly more efficient way; for smooth, regular problems the gain may be small or moderate.

Optimal grids for local error control are also characterized, and numerical examples show that current computational methods come close to generating optimal grids with respect to local error control. These simple examples also show that adaptivity is necessary in order to solve the problems numerically with a reasonable computational effort.

References

[1] C. Arévalo, J.D. López and G. Söderlind, *Linear multistep methods with constant coefficients and step density control*. To appear in J. Comp. and Appl. Math. (2006).

[2] U. Ascher, *Collocation for two-point boundary value problems revisited*, SIAM J. Numer. Anal. 23(3) (1986), 596–609.

[3] U. Ascher , J. Christiansen, and R.D. Russell, *COLSYS: Collocation Software or Boundary-Value ODEs*, ACM Transactions on Mathematical Software 7(2) (1981) 223–229.

[4] U. Ascher, R.M. Mattheij, and R.D. Russell, *The Numerical Solution of Two-point Boundary Value Problems*, Prentice Hall, Englewood Cliffs, New York, 1988.

[5] W. Auzinger, O. Koch and E.B. Weinmüller. *Efficient mesh selection for collocation methods applied to singular BVPs*, J. Comp. Appl. Math. **180** (2005), 213–227.

[6] B. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica 10 (2001), 1–102.

[7] J.R. Cash, *A survey of some global methods for solving two-point BVPs*, Appl. Numer. Anal. Comput. Math. 1(1-2) (2004) 7–17.

[8] C.C. Christara and K.S. Ng, *Adaptive techniques for spline collocation*, Computing (2005), Published online November 15, 2005.

[9] R.M. Corless, *A new view of the computational complexity of IVP for ODE*, Numerical Algorithms 31 (2002), 115–124.

[10] R.M. Corless, *An elementary solution of a minimax problem arising for automatic mesh selection*. SIGSAM Bulletin: Commun. Comput. Algebra 34(4) (2001), 7–15.

[11] E. Hairer and G. Söderlind. *Explicit, time reversible, adaptive step size control*. SIAM J. Sci. Comput., SISC 26 (2005) 1838–1851.

[12] G. Hardy, J.E. Littlewood, and J.E. Pólya. *Inequalities*. Cambridge University Press, 1952.

[13] S. Ilie. *Computational complexity of numerical solutions of initial value problems for differential algebraic equations*. PhD thesis, University of Western Ontario, 2005.

[14] S. Ilie, R.M. Corless, and G. Reid. *Numerical solutions of index-1 differential algebraic equations can be computed in polynomial time*, Numerical Algorithms 41(2) (2006) 161 – 171

[15] A. Iserles, *On the global error of discretization methods for highly-oscillatory ordinary differential equations*. BIT 42(3) (2002) 561 – 599.

[16] A. Iserles and G. Söderlind. *Global bounds on numerical error for ordinary differential equations*. J. Complexity 9 (1993) 97–112.

[17] P. Morino, R.H. Nochetto, and K.G. Siebert, *Convergence of adaptive finite element methods*, SIAM Review 44(4) (2002) 631–658.

[18] J. Niesen. *On the global error of discretization methods for ordinary differential equations*. PhD thesis, University of Cambridge, 2004.

[19] G. Pulverer, G. Söderlind, and E. Weinmüller. *Automatic grid control in adaptive BVP solvers*. To appear, 2007.

[20] R.D. Skeel, *Thirteen ways to estimate global error*. Numer. Math. 48(1) (1986) 1 – 20.

[21] G. Söderlind. *Digital filters in adaptive time-stepping*. ACM Trans. Math. Software 29 (2003), 1–26.

[22] G. Söderlind. *Time-step selection algorithms: adaptivity, control, and signal processing*. Appl. Numer. Math. 56 (3-4) (2006), 488–502.

[23] D. Viswanath, *Global errors of numerical ODE solvers and Lyapunov's theory of stability*. IMA J. Numer. Anal. 21(1) (2001) 387 – 406.

[24] A.G. Werschulz. *The Computational Complexity of Differential and Integral Equations*. Oxford Science Publications, 1991.