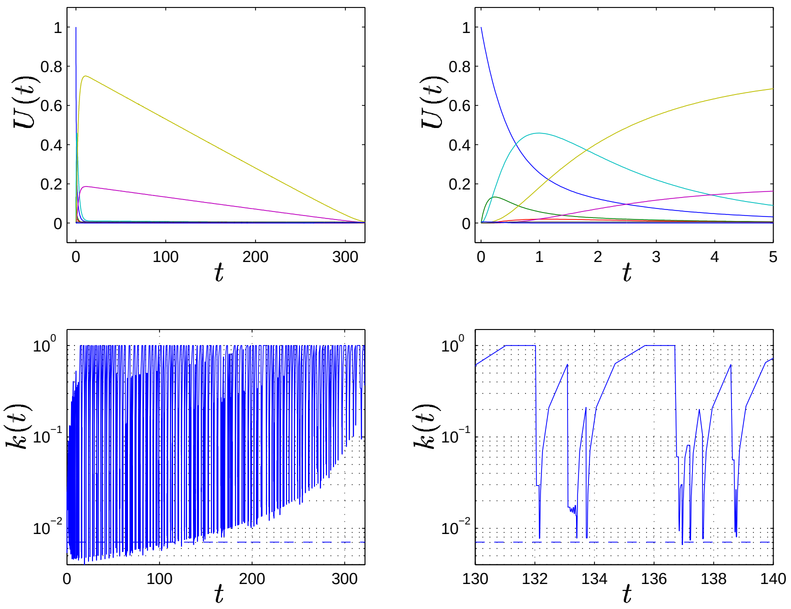
CHALMERS
FINITE ELEMENT CENTER



PREPRINT 2002–07

Explicit time-stepping for stiff ODEs

Kenneth Eriksson, Claes Johnson and Anders Logg

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2002

CHALMERS FINITE ELEMENT CENTER

Preprint 2002–07

Explicit time-stepping for stiff ODEs

Kenneth Eriksson, Claes Johnson and Anders Logg

CHALMERS

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, November 2002

Explicit time-stepping for stiff ODEs
Kenneth Eriksson, Claes Johnson and Anders Logg
NO 2002–07
ISSN 1404–4382

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 1000
Fax: +46 (0)31 772 3595
www.phi.chalmers.se

Printed in Sweden
Chalmers University of Technology
Göteborg, Sweden 2002

EXPLICIT TIME-STEPPING FOR STIFF ODES

KENNETH ERIKSSON, CLAES JOHNSON, AND ANDERS LOGG

Abstract. We present a new strategy for solving stiff ODEs with explicit methods. By adaptively taking a number of stabilising small explicit time steps when necessary, a stiff ODE system can be stabilised enough to allow for time steps much larger than what is indicated by classical stability analysis. For many stiff problems the cost of the stabilising small time steps is small and so the improvement is large. We illustrate the technique on a number of well-known stiff test problems.

1. Introduction

The classical wisdom developed in the 1950s regarding stiff ODEs is that efficient integration requires implicit (A-stable) methods, at least outside transients where the time steps may be chosen large from accuracy point of view. Using an explicit method (with a bounded stability region) the time steps have to be small at all times for stability reasons, in particular outside transients, and the advantage of a low cost per time step for the explicit method is counter-balanced by the necessity of taking a large number of small time steps. As a result the overall efficiency of an explicit method for a stiff ODE is small.

The question is now if it is possible to combine the low cost per time step of an explicit method with the possibility of choosing large time steps outside transients. To reach this favourable combination some kind of "stabilisation" of the explicit method seems to be needed, and the basic question is then if the stabilisation can be realized at a low cost.

We have been led to this question in our work on multi-adaptive cG(q) or dG(q) ODE-solvers based on Galerkin's method with continuous or discontinuous polynomials of degree q where individual time steps are used for different components, see [7, 8, 9, 10]. These methods are implicit and to realize efficient implementations, we need to use fixed point iteration with simple preconditioners, typically corresponding to diagonal scaling. With a limited (small) number of iterations, these iterative solvers correspond to explicit time-stepping and the same question of the cost of stabilisation arises.

When taking a large time step K with the explicit Euler method, modes of frequency λ with $\lambda K > 2$ will get amplified with an amplification factor $\lambda K - 1$. These modes may be damped by time-stepping with a small time step k satisfying $\lambda k < 2$ with a corresponding reduction factor $(1 - \lambda k)$ in each step. If $\lambda k \approx 1$ then the reduction is strong and a few small steps will be sufficient to kill the perturbation growth of the large time step. There

Date: November 28, 2002.

Key words and phrases. Stiff ODE, explicit methods, explicit Euler, Galerkin methods.

Department of Computational Mathematics, Chalmers University of Technology, S–412 96 Göteborg, Sweden, *email*: kenneth@math.chalmers.se, claes@math.chalmers.se, logg@math.chalmers.se.

is thus hope that an explicit method, such as e.g. the explicit Euler method, with large time steps may be used for stiff problems if large time steps are suitably accompanied by small time steps.

In this note we test this idea in adaptive form where the size and number of the small time steps are adaptively chosen according to the size of different residuals. In particular, we compute rates of divergence to determine the current mode λ of largest amplification and determine a corresponding small time step $k \approx \frac{1}{\lambda}$ with high reduction factor. We test the corresponding adaptive method in the setting of the cG(1) method with fixed point iteration, effectively corresponding to an explicit method if the number of iterations is kept small. We show in a sequence of test problems that the proposed adaptive method gives a significant reduction as compared to a standard implementation of an explicit method with the usual time step restriction. In these test problems we show efficiency gain factors with the new adaptive explicit method ranging from about 10 to 100. We conclude that for many stiff problems, we may efficiently use an explicit method, if only the explicit method is adaptively stabilised with a relatively small number of small time steps. We thus reach the desired combination of a low cost per time step and the possibility of taking large time steps outside transients.

Restricting the iterative method to simple fixed point iteration, possibly including diagonal scaling, implies that we have to rely on the inherent stability feature of the stiff ODE with rapid damping of high frequencies. A couple of well chosen small time steps may be sufficient for stabilisation, following the dynamics of the stiff ODE. This reasoning also indicates a limitation of this method of "natural damping": if the time step is small compared to the time scale of the component to be damped, then a large number of small steps will be needed and the efficiency will suffer. This typically happens when there is no clear gap between large and small eigenvalues of the linearised problem, since then the small damping time step will have to be small enough to match the dynamics of eigenmodes corresponding to large eigenvalues, which is not suitable for damping eigenmodes corresponding to medium sized eigenvalues.

We speculate that similar techniques may be used also to stabilise multi-grid smoothers.

2. The test equation

We consider first the *test equation*: Find $u : [0, \infty) \to \mathbb{R}$ such that

$$\begin{aligned} \dot{u}(t) + \lambda u(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{2.1}$$

where $\lambda > 0$ and u^0 is a given initial condition. The solution is given by $u(t) = \exp(-\lambda t)u^0$. We define the *transient* as $\{t > 0 : \lambda t \leq C\}$ with C a moderate constant. Outside the transient $u(t) = \exp(-\lambda t)u^0$ will be small.

The *implicit Euler method* for the test equation takes the form

$$U^n + k_n \lambda U^n = U^{n-1}, \quad \text{that is } U^n = (1 + k_n\lambda)^{-1} U^{n-1}.$$

This method is unconditionally stable (A-stable) and can be used with time steps k_n of arbitrary size. In the system case the implicit Euler method requires the solution of a system of equations.

Given a tolerance TOL > 0 for the error, the time step $k(t)$ at time t in the implicit Euler method should be chosen according to

$$k(t)|\dot{u}(t)| = \mathrm{TOL},$$

that is

$$k(t) = \frac{\exp(\lambda t)}{\lambda}\mathrm{TOL}.$$

The total number of time steps over the time interval $[0, T]$ is then

$$\int_0^T \frac{dt}{k(t)} \approx (\mathrm{TOL})^{-1},$$

independent of T and λ!

The *explicit Euler method* for the test equation reads

$$U^n = -k_n\lambda U^{n-1} + U^{n-1} = (1 - k_n\lambda)U^{n-1}.$$

This method is conditionally stable and requires that $k_n\lambda \leq 2$, which outside transients is too restrictive. We may view an explicit Euler step as an approximation to an implicit Euler step, where the explicit Euler step does not require the solution of a system of equations.

Now let K be a large time step satisfying $K\lambda > 2$ and let k a small time step chosen so that $k\lambda < 2$. Consider the method

$$U^n = (1 - k\lambda)^m(1 - K\lambda)U^{n-1}, \tag{2.2}$$

corresponding to one explicit Euler step with large time step K and m explicit Euler steps with small time steps k, where m is a positive integer to be determined. Altogether this corresponds to a time step of size $k_n = K + mk$. Defining the polynomial function $p(x) = (1 - \theta x)^m(1 - x)$, where $\theta = \frac{k}{K}$, we can write the method (2.2) in the form

$$U^n = p(K\lambda)U^{n-1}.$$

We now seek to choose m so that

$$|p(K\lambda)| \leq 1,$$

which is needed for stability. We need to satisfy

$$|1 - k\lambda|^m(K\lambda - 1) \leq 1,$$

that is

$$m \geq \frac{\log(K\lambda - 1)}{-\log|1 - k\lambda|} \approx \frac{\log(K\lambda)}{c}, \tag{2.3}$$

with $c = k\lambda$ a moderate constant, for definiteness $c = 1/2$.

We conclude that m will be quite small and only a small fraction of the total time will be spent on time-stepping with the small time step k. To see this, define the *cost* as

$$\alpha = \frac{1+m}{K+km} \in (1/K, 1/k),$$

i.e. the number of time steps per unit interval. The classical stability analysis gives

$$\alpha = 1/k = \lambda/2, \tag{2.4}$$

with a maximum time step $k = 2/\lambda$. Using (2.3) we instead find

$$\alpha \approx \frac{1+\log(K\lambda)/c}{K+\log(K\lambda)/\lambda} \approx \frac{\lambda}{c}\log(K\lambda)/(K\lambda) \ll \lambda/c, \tag{2.5}$$

for $K\lambda \gg 1$. The cost is thus decreased by the cost reduction factor

$$\frac{2\log(K\lambda)}{cK\lambda} \sim \frac{\log(K\lambda)}{K\lambda},$$

which can be quite significant for large values of $K\lambda$.

Choosing the small time step k and the number of small time steps m corresponds to positioning the zeros of the polynomial p. In [12] explicit methods with extended stability regions are constructed using shifted Chebyshev polynomials, and the resulting polynomials have similarities with the polynomial p constructed above.

3. The test system

We consider now the *test system*: Find $u : [0,\infty) \to \mathbb{R}$ such that

$$\begin{aligned} \dot{u}(t) + Au(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{3.1}$$

where $A = \mathrm{diag}(\lambda_i)$ is a diagonal $N \times N$ matrix with diagonal elements $0 \leq \lambda_1 \leq ... \leq \lambda_N$.

The *implicit Euler method* for the test system takes the form

$$U^n + k_n A U^n = U^{n-1}, \quad \text{that is } U^n = (I + k_n A)^{-1} U^{n-1}.$$

This method is unconditionally stable (A-stable) and can be used with time steps k_n of arbitrary size.

The *explicit Euler method* for the test system reads

$$U^n = -k_n A U^{n-1} + U^{n-1} = (I - k_n A) U^{n-1}.$$

This method is conditionally stable and requires that $k_n \lambda_N \leq 2$, which outside transients is too restrictive.

Let now K be a large time step satisfying $K\lambda_N > 2$ and let k a small time step chosen so that $k\lambda_N < 2$, for definiteness $k\lambda_N = c = 1/2$. Consider the method

$$U^n = (I - kA)^m (I - KA) U^{n-1}, \tag{3.2}$$

corresponding to one explicit Euler step with large time step K and m explicit Euler steps with small time steps k, where m is a positive integer to be determined. Altogether this corresponds to a time step of size $k_n = K + mk$. With $p(x) = (1-\theta x)^m(1-x)$ as before, where $\theta = \frac{k}{K}$, we can write the method (3.2) in the form

$$U^n = p(KA)U^{n-1}.$$

We now seek to choose m so that for $i = 1, ..., N$,

$$|p(K\lambda_i)| \leq 1$$

which is needed for stability. In particular we need to satisfy

$$|1 - k\lambda_i|^m (K\lambda_i - 1) \leq 1,$$

for all i with $K\lambda_i > 2$, that is

$$m \geq \frac{\log(K\lambda_i - 1)}{-\log|1 - k\lambda_i|} \approx \frac{K}{k}\frac{\log(K\lambda_i - 1)}{K\lambda_i}.$$

Taking the minimum value of m for the worst-case λ_i, we find

$$m \approx K/3k.$$

We conclude that $mk \approx K/3$ and one fourth of the time we will need to use small time steps. The cost reduction factor is now not smaller than

$$\frac{\alpha}{\lambda_N/2} \approx \frac{2}{c}\frac{k+mk}{K+mk} \approx 1.$$

This inefficiency comes from the fact that $k\lambda_i$ may be small, although $K\lambda_i \gg 2$, and thus also the damping of the factor $1 - k\lambda_i$. This typically happens when we have a range of eigenvalues in a large interval. If however we have only a couple of large eigenvalues, say $K\lambda_N \geq K\lambda_{N-1} \gg 2$, while $K\lambda_i \leq 2$ for $i \leq N$, we need

$$m \gtrsim \frac{K}{k}\frac{\log(K\lambda_{N-1})}{K\lambda_{N-1}},$$

and thus $mk \approx K\log(K\lambda_{N-1})/(K\lambda_{N-1})$, which gives a cost reduction factor of

$$\frac{\alpha}{\lambda_N/2} \approx \frac{2\log(K\lambda_{N-1})}{cK\lambda_{N-1}} \sim \frac{\log(K\lambda_{N-1})}{K\lambda_{N-1}},$$

which can still be quite substantial if $K\lambda_{N-1} \gg 1$. We conclude that if the large eigenvalues are well separated from the smaller eigenvalues, we can damp out the corresponding eigenmodes with a number of small stabilising time steps in an efficient way.

4. The general non-linear problem

We consider now the general non-linear problem,

$$\begin{aligned} \dot{u}(t) &= f(u(t),t), \;\; t \in (0,T], \\ u(0) &= u^0, \end{aligned} \tag{4.1}$$

where $u : [0, T] \to \mathbb{R}^N$, $f : \mathbb{R}^N \times (0, T] \to \mathbb{R}^N$ is a given bounded and differentiable function and $T > 0$ a given final time. The explicit Euler method for (4.1) reads

$$U^n = U^{n-1} + k_n f(U^{n-1}, t_{n-1}),$$

where $0 = t_0 < t_1 < \ldots < t_M = T$ is a partition of $[0, T]$ and the function $U(t)$ is piecewise constant and left-continuous with $U^n = U(t_n^+)$. The exact solution u satisfies a similar relation:

$$u^n = u^{n-1} + \int_{t_{n-1}}^{t_n} f(u(t), t)\ dt,$$

where $u^n = u(t_n)$. The error $e(t) = U(t) - u(t)$ then satisfies

$$\begin{aligned} e(t_n^+) - e(t_{n-1}^+) &= (U^n - u^n) - (U^{n-1} - u^{n-1}) = (U^n - U^{n-1}) - (u^n - u^{n-1}) \\ &= \int_{t_{n-1}}^{t_n} (f(U^{n-1}, t_{n-1}) - f(u(t), t))\ dt \\ &\approx \int_{t_{n-1}}^{t_n} (f(U^{n-1}, t) - f(u(t), t))\ dt = \int_{t_{n-1}}^{t_n} J(\xi(t), t)(U(t) - u(t))\ dt \\ &= \int_{t_{n-1}}^{t_n} Je\ dt, \end{aligned}$$

where J is the Jacobian $\frac{\partial f}{\partial u}$ of the right-hand side evaluated at a mean value of U and u and we have neglected a term of size $k_n^2|\frac{\partial f}{\partial t}|$. If the problem is autonomous this term is zero, but if $|\frac{\partial f}{\partial t}|$ is large this term is not negligible. We comment more on this below.

If now J is diagonalisable, i.e. there exists an orthogonal matrix $X(t)$ such that we can write $X^t(t)J(t)X(t) = \Lambda(t) = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$, we set $e(t) = X(t)v(t)$. If also the time step is small enough so that J is piecewise constant with good approximation (or f is linear), we have

$$v(t_n^+) - v(t_{n-1}^+) = \int_{t_{n-1}}^{t_n} \Lambda v\ dt,$$

which for $v|_{(t_{n-1}, t_n)}$ monotone gives the estimate

$$v_i(t_n^+) \in ((1 + k_n\lambda_i)v_i(t_{n-1}^+), v_i(t_{n-1}^+)/(1 - k_n\lambda_i)),$$

$i = 1, \ldots, N$. We conclude that with a suitable choice of time step k_n we can damp out stiff eigenmodes in the error corresponding to $\lambda_i \ll 0$ with a small number of explicit Euler steps as in Sections 2 and 3.

This simple analysis also shows that the simple strategy may not work for non-autonomous problems if $|\frac{\partial f}{\partial t}|$ is large. In that case we are in some way always in a transient, since in every time step there will be a perturbation corresponding to $k_n\frac{\partial f}{\partial t}$, see Section 8.8 below.

5. Iterative methods

From another viewpoint, we may consider using an explicit-type iterative method for solving the discrete equations arising from an implicit method. The implicit cG(1) method with midpoint quadrature for the general non-linear problem (4.1) reads

$$U^n = U^{n-1} + k_n f(\frac{U^{n-1} + U^n}{2}, \frac{t_{n-1} + t_n}{2}). \tag{5.1}$$

We can solve this system of non-linear equations for U^n using Newton's method, but the simplest and cheapest method is to apply fixed point iteration directly to (5.1), i.e.

$$U^{nl} = U^{n-1} + k_n f(\frac{U^{n-1} + U^{n,l-1}}{2}, \frac{t_{n-1} + t_n}{2}),$$

for $l = 0, 1, \dots$ until convergence with $U^{n,0} = U^{n-1}$. The fixed point iteration converges for small enough time step k_n, and so the stability condition for a standard explicit method appears also in explicit-type iterative methods as a condition for convergence of the iterative solution of the implicit equations.

To determine a stop criterion for the fixed point iteration, we measure the size of the *discrete residual*,

$$r^{nl} = \frac{1}{k_n}(U^{nl} - U^{n-1}) - f(\frac{U^{n-1} + U^{nl}}{2}, \frac{t_{n-1} + t_n}{2}),$$

which should be zero for the true cG(1) approximation. We continue the iterations until the discrete residual is smaller than some tolerance tol > 0. Usually only a couple of iterations are needed. Estimating the error $e(T) = U(T) - u(T)$ at final time T (see [7]), we have

$$\|e(T)\| \leq S(T) \max_{[0,T]} k\|R\| + S^0(T) \max_{[0,T]} \|r\|,$$

where $R(t) = \dot{U}(t) - f(U(t), t)$ is the *continuous residual* and $S(T)$ and $S^0(T)$ are stability factors. For the test equation we have $S(T) \leq 1$ and $S^0(T) \leq 1/\lambda$, which suggests that for a typical stiff problem we can take tol $=$ TOL, where TOL is a tolerance for the error $e(T)$ at final time.

The discrete residual satisfies the following simple relation:

$$\begin{aligned} r^{nl} &= \tfrac{1}{k_n}(U^{nl} - U^{n-1}) - f(\tfrac{U^{n-1}+U^{nl}}{2}, \tfrac{t_{n-1}+t_n}{2}) \\ &= f(\tfrac{U^{n-1}+U^{n,l-1}}{2}, \tfrac{t_{n-1}+t_n}{2}) - f(\tfrac{U^{n-1}+U^{nl}}{2}, \tfrac{t_{n-1}+t_n}{2}) \\ &= J(\xi_l)\tfrac{U^{n,l-1}-U^{nl}}{2} = \tfrac{1}{2}J(\xi_l)(U^{n,l-1} - U^{n-1} - k_n f(\tfrac{U^{n-1}+U^{n,l-1}}{2}, \tfrac{t_{n-1}+t_n}{2})), \end{aligned}$$

which gives

$$r^{nl} = \frac{k_n}{2} J(\xi_l) r^{n,l-1}. \tag{5.2}$$

By measuring the size of the discrete residual, we thus obtain valuable information about stiff eigenmodes that we need to take care of with the small damping steps. Notice now that if we are able to damp out completely the eigenmode corresponding to the largest (negative) eigenvalue, the eigenmode corresponding to the next largest eigenvalue will reveal itself in the iterative process, enabling us to choose a new small time step (but somewhat larger) targeted at damping out the next largest stiff eigenmode. This also means that if we have only a couple of large eigenvalues and we have managed to damp out the corresponding eigenmodes with small damping steps, the fixed point iteration will seem to converge since we are in practice iterating only on eigenmodes corresponding to the remaining small eigenvalues. If we continue with the iterations, the eigenmodes corresponding to large eigenvalues will appear again and the iteration will start to diverge, but if we have reached the stop criterion before that, we accept the solution and move on to the next time step. If then later at some time step, the bad eigenmodes have grown again so that the iterations don't converge, we measure the size of the discrete residual and take a number of small damping steps to take care of the bad eigenmodes. We discuss this in more detail below in Section 7.

6. Multi-adaptive solvers

In a multi-adaptive solver we use individual time steps for different components. A crucial part of the algorithm described in [7, 8] is the iterative fixed point solution of the discrete equations on time slabs. The simple strategy to take small damping steps to stabilise the system applies also in the multi-adaptive setting, where we may also target individual eigenmodes (if these are represented by different components) using individual damping steps. We explore this further in [10].

7. An adaptive algorithm

The question is now if we can choose the time step sequence automatically in an adaptive algorithm. We approach this question in the setting of an implicit method combined with an explicit-type iterative solver as in Section 5.

A simple adaptive algorithm for the standard cG(1) method with iterative solution of the discrete equations reads:

(1) Determine a suitable initial time step k_1 and solve the discrete equations for the solution on $U(t)$ on (t_0, t_1).
(2) Repeat on (t_{n-1}, t_n) for $n = 2, 3, \dots$ until $t_n \geq T$:
    (a) Evaluate the continuous residual R_{n-1} from the previous time interval.
    (b) Determine the new time step k_n based on R_{n-1}.
    (c) Solve the discrete equations on (t_{n-1}, t_n) using fixed point iteration.

In reality we want to control the global error, which means we have to solve also the dual problem, compute stability factors (or weights), evaluate an a posteriori error estimate and possibly repeat the process until the error is below a given tolerance TOL > 0. The full algorithm is thus slightly more elaborate, but the basic algorithm presented here is the central part. See [3] for a discussion.

We comment also on step (2.b). For the cG(1) method we would like to take $k_n = \mathrm{TOL}/(S \cdot \|R_{n-1}\|)$ but this introduces unwanted oscillations in the size of the time step. A small residual gives a large time step which results in a large residual, and so on. To avoid this, the simple step size selection has to be combined with a regulator of some kind, see [4, 11] or [8]. It turns out that a simple strategy that works well in many situations is to take k_n as the geometric mean

$$k_n = \frac{2\tilde{k}_n k_{n-1}}{\tilde{k}_n + k_{n-1}},$$

where $\tilde{k}_n = \mathrm{TOL}/(S \cdot \|R_{n-1}\|)$.

Now for a stiff problem, what may go wrong is step (2.c); if the time step k_n is too large the fixed point iterations will not converge. To be able to handle stiff problems using the technique discussed above, we propose the following simple algorithm:

(1) Determine a suitable initial time step k_1 and solve the discrete equations for the solution $U(t)$ on (t_0, t_1).

(2) Repeat on (t_{n-1}, t_n) for $n = 2, 3, \ldots$ until $t_n \geq T$:

- (a) Evaluate the continuous residual R_{n-1} from the previous time interval.
- (b) Determine the new time step k_n based on R_{n-1}.
- (c) Solve the discrete equations on (t_{n-1}, t_n) using fixed point iteration.
- (d) If (2.c) didn't work, compute

$$L = \frac{2}{k_n} \frac{\|r^l\|}{\|r^{l-1}\|},$$

  and do $m = \log(k_n L)$ explicit Euler steps with $k = c/L$ and c close to 1.
- (e) Try again starting at (2.a) with $n \to n + m$.

In the analysis of Sections 2 and 3 we had $c = 1/2$, but it is clear that the damping steps will be more efficient if we have c close to 1. If we take $c = 1$ the bad eigenmode will be damped out completely (at least for the simple test equation) which we may not want. A suitable value is $c = 0.99$. An implementation of this algorithm in the form of a simple MATLAB code is available for inspection [1], including the test problems presented in the next section.

8. Examples

To illustrate the technique, we take a simple standard implementation of the cG(1)-method (with explicit fixed point solution of the discrete equations) and add a couple of lines to handle the damping for stiff problems. We try this code on a number of well-known stiff problems taken from the ODE literature, and conclude that we are able to handle stiff problems with this explicit code.

When referring to the cost α below, this includes also the number of fixed point iterations needed to compute the cG(1) solution on intervals where the iterations converge. This is compared to the cost α_0 for the standard cG(1) method in which we are forced to take small time steps all the time. (These small time steps are marked by dotted lines.) For all example problems below we report both the cost α and the cost reduction factor α/α_0.

8.1. **The test equation.** The first problem we try is the test equation:

$$\begin{aligned} \dot{u}(t) + \lambda u(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{8.1}$$

on $[0, 10]$, where we choose $u^0 = 1$ and $\lambda = 1000$. As is evident from Figure 1, the time step sequence is automatically chosen in agreement with the previous discussion. The cost is only $\alpha \approx 6$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/310$.

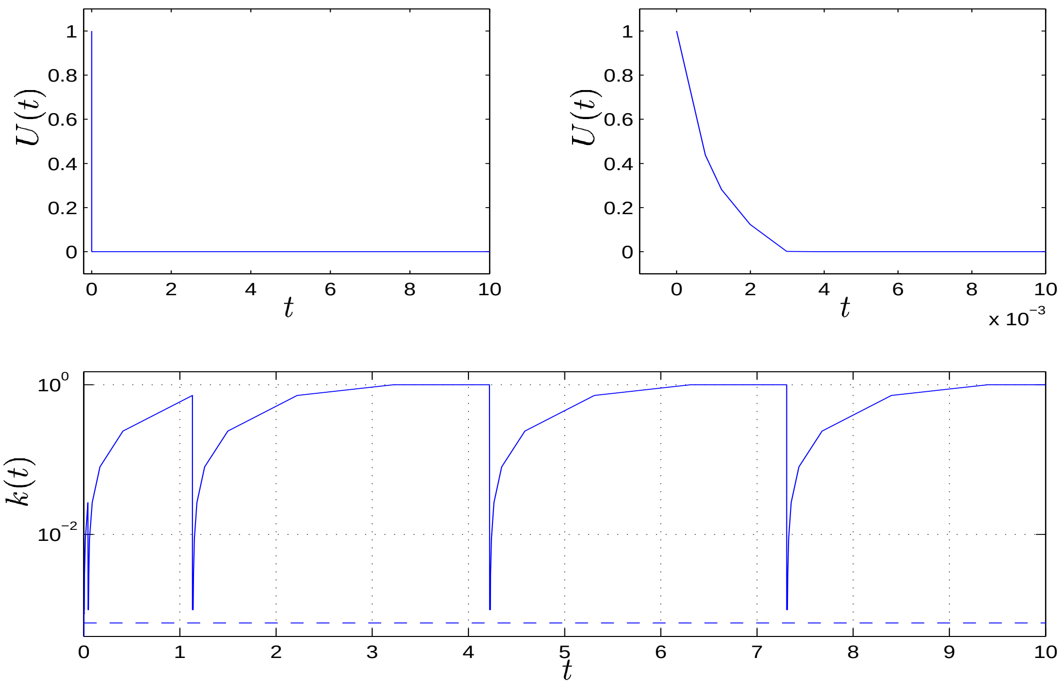


FIGURE 1. Solution and time step sequence for eq. (8.1), $\alpha/\alpha_0 \approx 1/310$.

8.2. **The test system.** For the test system,

$$\begin{aligned} \dot{u}(t) + Au(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{8.2}$$

on $[0, 10]$, we take $A = \operatorname{diag}(100, 1000)$ and $u^0 = (1, 1)$. There are now two eigenmodes with large eigenvalues that have to be damped out. The dominant eigenvalue of A is $\lambda_2 = 1000$ and most of the damping steps are chosen to damp out this eigenmode, but some of the damping steps are chosen based on the second largest eigenvalue $\lambda_1 = 100$. When to damp out which eigenmode is automatically decided by the adaptive algorithm; the bad eigenmode that needs to be damped out becomes visible in the iterative solution process. Since there is an additional eigenvalue, the cost is somewhat larger than for the scalar test problem, $\alpha \approx 18$, giving a cost reduction factor of $\alpha/\alpha_0 \approx 1/104$.

8.3. **A problem with three scales.** As another simple test problem we consider the test system again,

$$\begin{aligned} \dot{u}(t) + Au(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{8.3}$$

on $[0, 10]$, this time with $A = \operatorname{diag}(10, 100, 1000)$, and $u^0 = (1, 1, 1)$. Even if we now have three scales the cost is the same as for the previous problem: $\alpha \approx 18$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/107$.

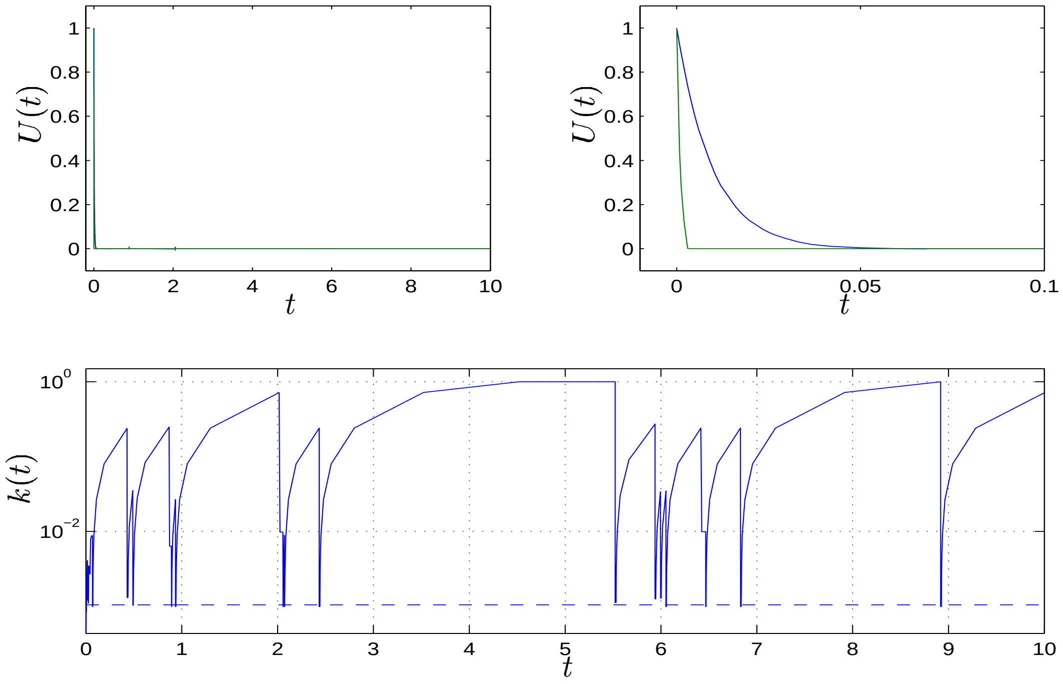


FIGURE 2. Solution and time step sequence for eq. (8.2), $\alpha/\alpha_0 \approx 1/104$.

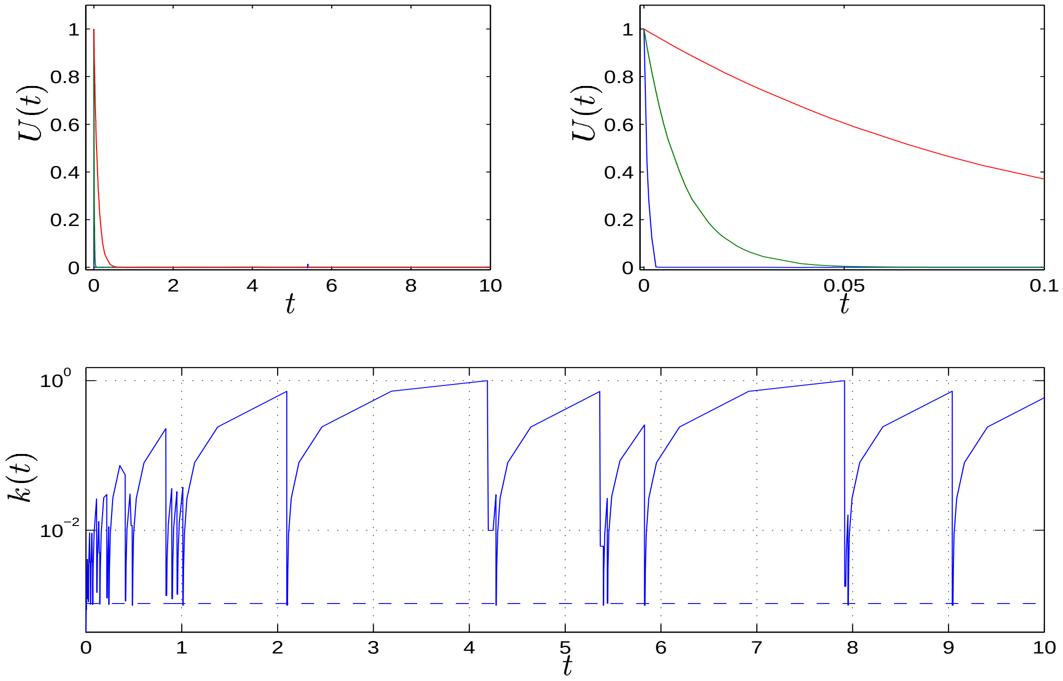


FIGURE 3. Solution and time step sequence for eq. (8.3), $\alpha/\alpha_0 \approx 1/107$.

8.4. **A linear non-normal problem.** The method behaves similarly to the simple test system even if we make the matrix A highly non-normal. We now solve

$$\begin{aligned} \dot{u}(t) + Au(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{8.4}$$

on $[0, 10]$, with

$$A = \begin{bmatrix} 1000 & -10000 \\ 0 & 100 \end{bmatrix},$$

and $u^0 = (1, 1)$. The cost is about the same as for the previous problem, $\alpha \approx 17$, but the cost reduction factor is smaller: $\alpha/\alpha_0 \approx 1/180$.

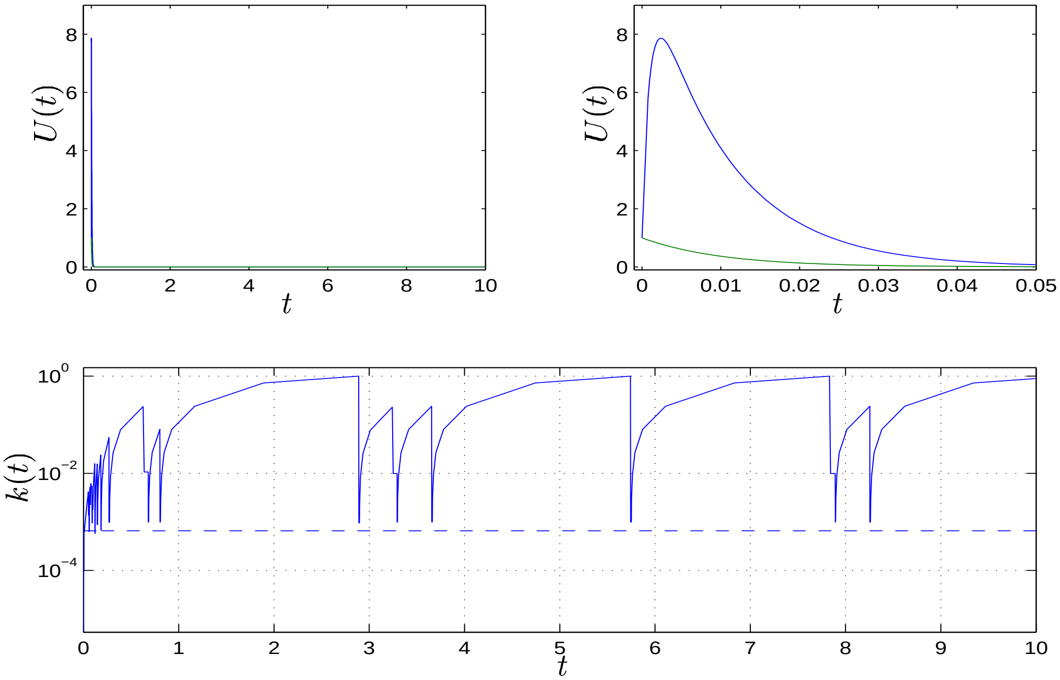


FIGURE 4. Solution and time step sequence for eq. (8.4), $\alpha/\alpha_0 \approx 1/180$.

8.5. **A chemical reaction system.** The next problem originating from 1966 (Robertson) is taken from the book on methods for stiff ODEs by Hairer and Wanner [5]. This problem models the following system of chemical reactions:

$$\begin{array}{cccl} A & \xrightarrow{0.04} & B & \text{(slow)} \\ B + B & \xrightarrow{3 \cdot 10^7} & B + C & \text{(very fast)} \\ B + C & \xrightarrow{10^4} & A + C & \text{(fast)} \end{array}$$

for which the equations are:

$$\left\{\begin{array}{rcl} \dot{u}_1 & = & -0.04u_1 + 10^4 u_2 u_3, \\ \dot{u}_2 & = & 0.04u_1 - 10^4 u_2 u_3 - 3\cdot 10^7 u_2^2, \\ \dot{u}_3 & = & 3\cdot 10^7 u_2^2. \end{array}\right. \tag{8.5}$$

We compute the solution on $[0, 0.3]$ with $u^0 = (1, 0, 0)$. The very large parameters of sizes 10^4 and 10^7 forces us to take quite small time steps and the cost is now $\alpha \approx 600$, giving a cost reduction factor which is not smaller than $\alpha/\alpha_0 \approx 1/5$.

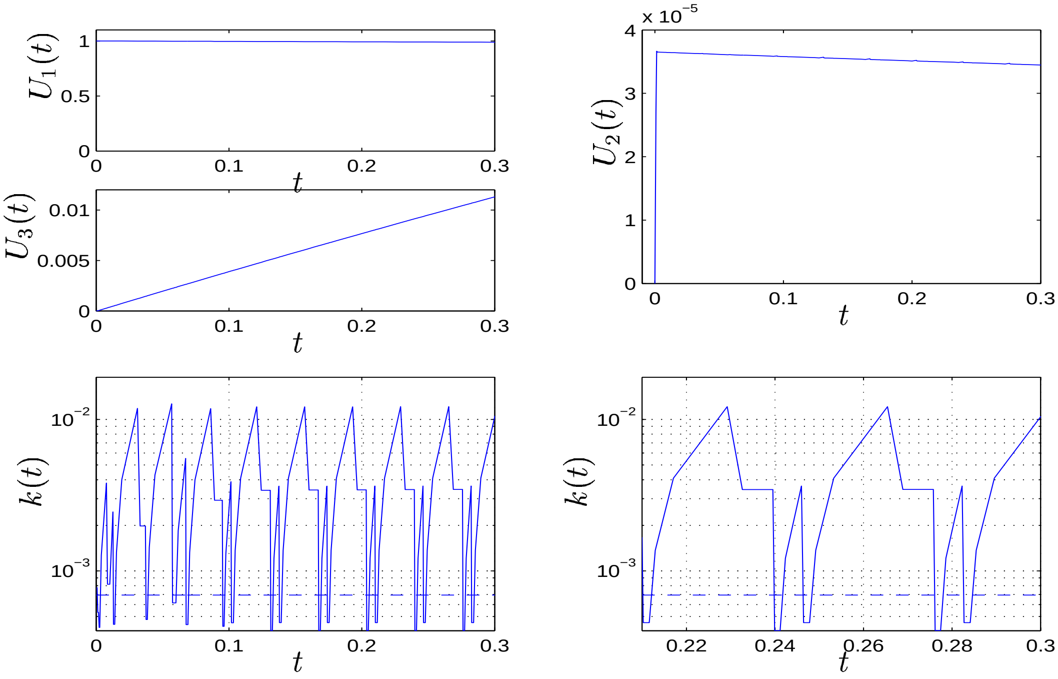


FIGURE 5. Solution and time step sequence for eq. (8.5), $\alpha/\alpha_0 \approx 1/5$.

8.6. **The HIRES problem.** The so-called HIRES problem ("High Irradiance RESponse") originates from plant physiology and is taken from the test set of ODE problems compiled by Lioen and de Swart [6]. The problem consists of the following eight equations:

$$\left\{\begin{array}{rcl} \dot{u}_1 & = & -1.71u_1 + 0.43u_2 + 8.32u_3 + 0.0007, \\ \dot{u}_2 & = & 1.71u_1 - 8.75u_2, \\ \dot{u}_3 & = & -10.03u_3 + 0.43u_4 + 0.035u_5, \\ \dot{u}_4 & = & 8.32u_2 + 1.71u_3 - 1.12u_4, \\ \dot{u}_5 & = & -1.745u_5 + 0.43u_6 + 0.43u_7, \\ \dot{u}_6 & = & -280.0u_6u_8 + 0.69u_4 + 1.71u_5 - 0.43u_6 + 0.69u_7, \\ \dot{u}_7 & = & 280.0u_6u_8 - 1.81u_7, \\ \dot{u}_8 & = & -280.0u_6u_8 + 1.81u_7, \end{array}\right. \tag{8.6}$$

together with the initial condition $u^0 = (1.0, 0, 0, 0, 0, 0, 0, 0.0057)$. We integrate over $[0, 321.8122]$ (as specified in [6]) and present the solution and the time step sequence in Figure 6. The cost is now $\alpha \approx 8$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/33$.

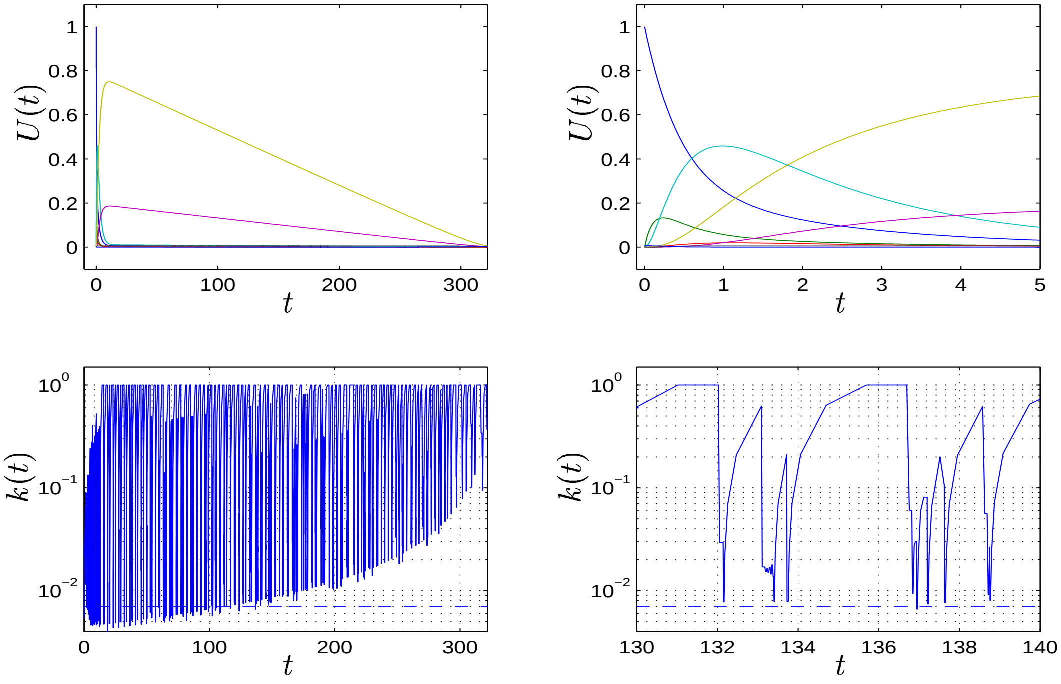


FIGURE 6. Solution and time step sequence for eq. (8.6), $\alpha/\alpha_0 \approx 1/33$.

8.7. **The Akzo-Nobel problem.** Another problem from the ODE test set [6] is the "Chemical Akzo-Nobel" problem, which consists of the following six equations:

$$
\left\{
\begin{array}{rcl}
\dot{u}_1 & = & -2r_1 + r_2 - r_3 - r_4, \\
\dot{u}_2 & = & -0.5r_1 - r_4 - 0.5r_5 + F, \\
\dot{u}_3 & = & r_1 - r_2 + r_3, \\
\dot{u}_4 & = & -r_2 + r_3 - 2r_4, \\
\dot{u}_5 & = & r_2 - r_3 + r_5, \\
\dot{u}_6 & = & -r_5,
\end{array}
\right.
\tag{8.7}
$$

where $F = 3.3 \cdot (0.9/737 - u_2)$ and the reaction rates are given by $r_1 = 18.7 \cdot u_1^4 \sqrt{u_2}$, $r_2 = 0.58 \cdot u_3 u_4$, $r_3 = 0.58/34.4 \cdot u_1 u_5$, $r_4 = 0.09 \cdot u_1 u_4^2$ and $r_5 = 0.42 \cdot u_6^2 \sqrt{u_2}$. We integrate over the interval $[0, 180]$ with initial condition $u^0 = (0.437, 0.00123, 0, 0, 0, 0.367)$. Allowing a maximum time step of $k_{\max} = 1$ (chosen arbitrarily), the cost is only $\alpha \approx 2$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/9$. The actual gain in a specific situation is determined by the quotient between the large time steps and the small damping time steps, as well as the number of small damping steps that are needed. In this case the number of small damping steps is small, but the large time steps are not very large compared to the small damping steps. The gain is thus determined both by the stiff nature of the problem and the tolerance (or the size of the maximum allowed time step).

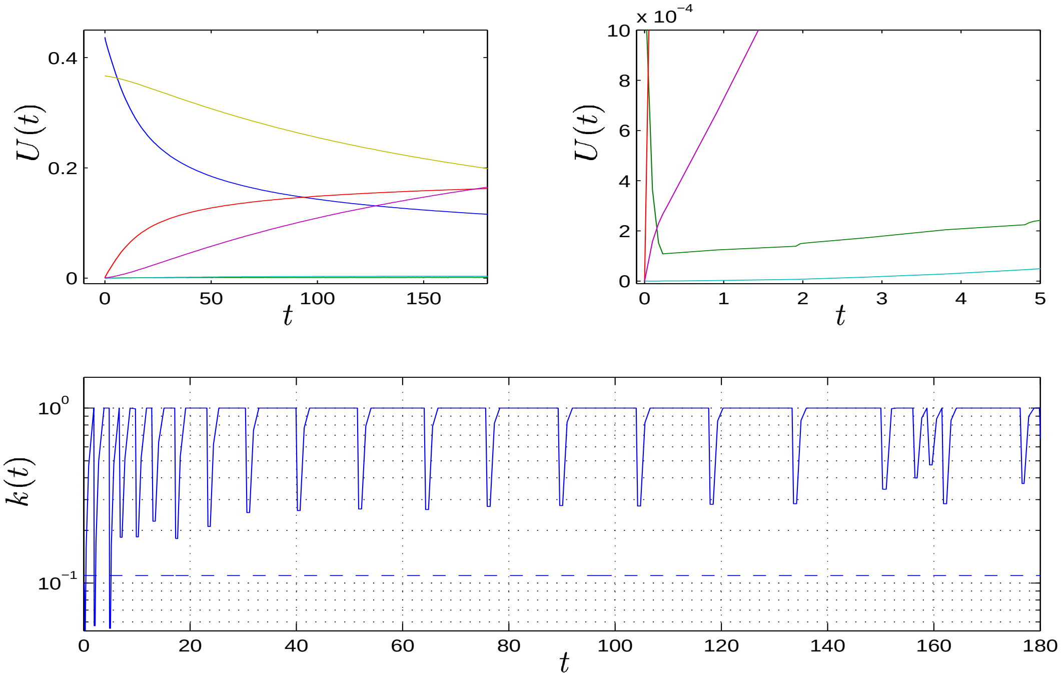


FIGURE 7. Solution and time step sequence for eq. (8.7), $\alpha/\alpha_0 \approx 1/9$.

8.8. **A non-autonomous problem.** The following problem is taken from the book by Ascher and Petzold [2] and is different from the previous problems in that the system is non-autonomous:

$$\dot{u}(t) = -100(u(t) - \sin(t)). \tag{8.8}$$

We compute the solution on $[0, 10]$ for $u^0 = 1$. The cost is now $\alpha \approx 100$ and the effective time step close to $1/100$. The cost reduction factor is now $\alpha/\alpha_0 \approx 2/3$. For this particular problem the simple damping strategy does not work, as discussed in Section 4, since with $\frac{\partial f}{\partial t} \sim 100$ a perturbation of size $100k_n$ will be introduced in every time step.

8.9. **Van der Pol's equation.** The second stiff problem in the book by Hairer and Wanner [5] is Van der Pol's equation:

$$\ddot{u} + \mu(u^2 - 1)\dot{u} + u = 0,$$

which we write as

$$\left\{ \begin{array}{rcl} \dot{u}_1 & = & u_2, \\ \dot{u}_2 & = & -\mu(u_1^2 - 1)u_2 - u_1. \end{array} \right. \tag{8.9}$$

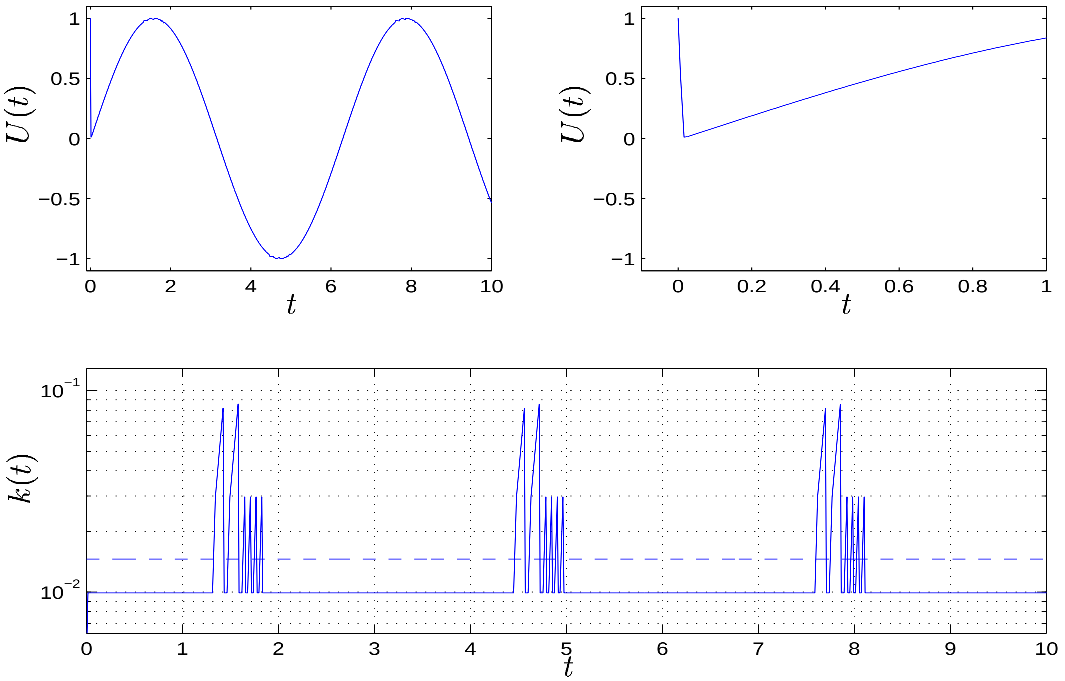


FIGURE 8. Solution and time step sequence for eq. (8.8), $\alpha/\alpha_0 \approx 2/3$.

We take $\mu = 1000$ and solve on the interval $[0, 10]$ with initial condition $u^0 = (2, 0)$. The time step sequence behaves as desired with only a small portion of the time spent on taking small damping steps. The cost is now $\alpha \approx 140$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/75$.

8.10. **The heat equation.** A special stiff problem is the one-dimensional heat equation:

$$\begin{aligned} \dot{u}(x,t) - u''(x,t) &= f(x,t), \quad x \in (0,1), t > 0, \\ u(0) = u(1) &= 0, \\ u(\cdot, t) &= 0, \end{aligned}$$

where we choose $f(x,t) = f(x)$ as an approximation of the Dirac delta function at $x = 0.5$. Discretising in space, we obtain the ODE

$$\begin{aligned} \dot{u}(t) + Au(t) &= f, \\ u(0) &= 0, \end{aligned} \tag{8.10}$$

where A is the *stiffness matrix*. With a spatial resolution of $h = 0.01$, the eigenvalues of A are distributed in the interval $[-4 \cdot 10^4, 0]$ (see Figure 10). Following the discussion in Section 3, we then expect to have only a moderate reduction factor since there is no clear separation into large and small eigenvalues. However, because the damping steps are more precise than assumed in the simple analysis of Section 3 (corresponding to $c = 0.99$ rather than $c = 0.5$) and also because the time step regulation mechanism gives a sequence

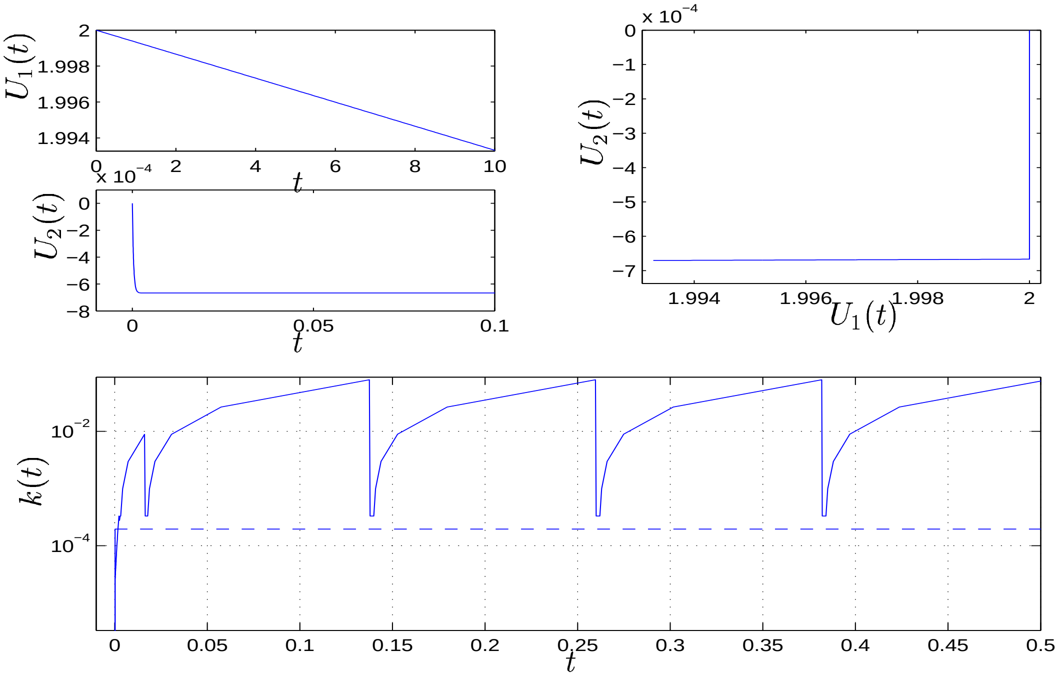


Figure 9. Solution and time step sequence for eq. (8.9), $\alpha/\alpha_0 \approx 1/75$.

of different time steps rather than just small time steps k and large time steps K, we are able to do a little better. The cost is now $\alpha \approx 4000$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/17$.

8.11. **A non-stiff problem.** To show that the method works equally well for non-stiff problems, we finally consider the following simple system:

$$\left\{ \begin{array}{rcl} \dot{u}_1 & = & 5u_2, \\ \dot{u}_2 & = & -u_1. \end{array} \right. \tag{8.11}$$

With initial condition $u^0 = (0, 1)$, the solution is $u(t) = (\sqrt{5}\sin(\sqrt{5}t), \cos(\sqrt{5}))$. Since this problem is non-stiff (for reasonable tolerances), the solver never needs to take any small damping steps and works as a standard non-stiff cG(1) solver with no overhead. This is also evident from the time step sequence (see Figure 10) which is chosen only to match the size of the (continuous) residual.

9. Conclusions

By taking a small number of stabilising time steps when needed, an explicit method can be extended to handle stiff problems efficiently. The actual gain in a specific situation (as compared to taking small time steps all the time) is determined by accuracy requirements and the nature of the problem. The problems most well suited for this technique are

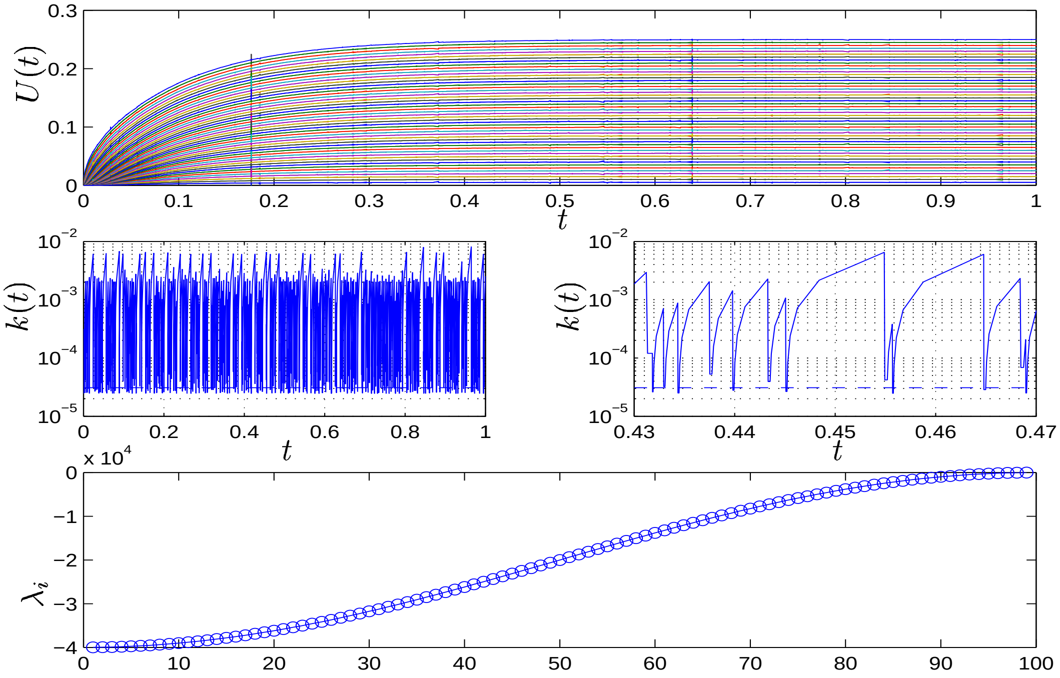


FIGURE 10. Solution and time step sequence for eq. (8.10), $\alpha/\alpha_0 \approx 1/17$.

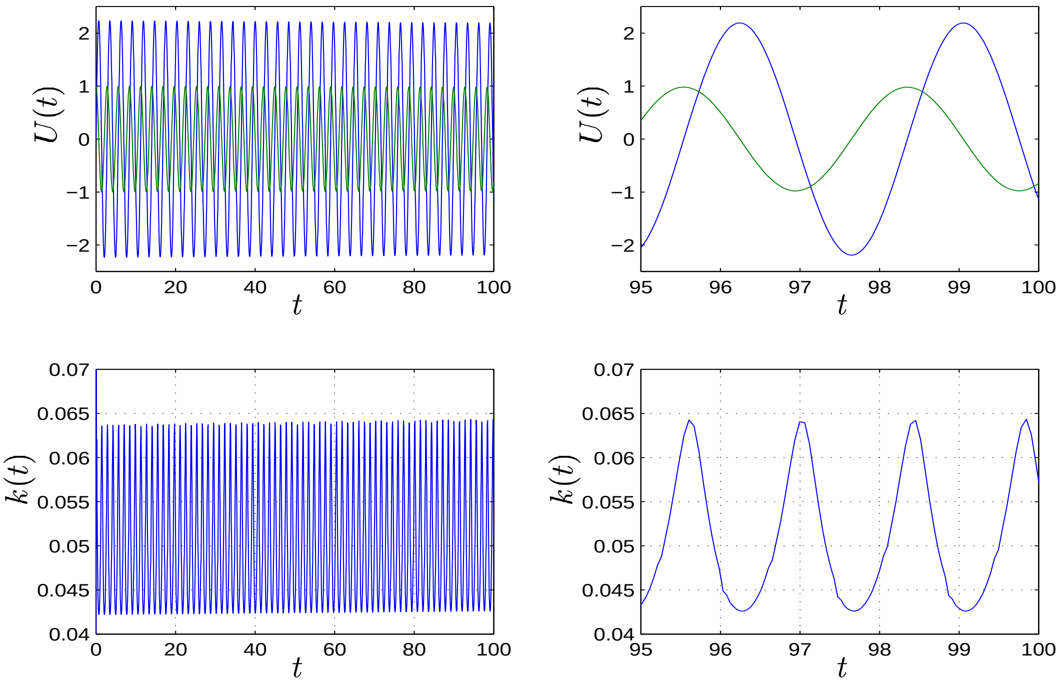


FIGURE 11. Solution and time step sequence for eq. (8.11), $\alpha/\alpha_0 = 1$.

problems where large eigenvalues of the Jacobian matrix are well separated from smaller eigenvalues.

This resolves a part of the mystery of the stiffness problem: problems with simple and smooth solutions and large damping should be easy to solve, not difficult. It is in fact the inherent damping of stiff problems that makes it possible to damp out the bad eigenmodes, with a couple of cheap and simple explicit Euler steps, and thus enables efficient solution of stiff problems using explicit methods.

References

[1] *Chalmers finite element center software*, `http://www.phi.chalmers.se/software/`, (2002).

[2] U. Ascher and L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, 1998.

[3] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, in Acta Numerica 1995, Cambridge University Press, 1995, pp. 105–158.

[4] K. Gustafsson, M. Lundh, and G. Söderlind, *A pi stepsize control for the numerical solution of ordinary differential equations*, BIT, 28 (1988), pp. 270–287.

[5] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II — Stiff and Differential-Algebraic Problems*, Springer Series in Computational Mathematics, vol 14, 1991.

[6] W. Lioen and H. de Swart, *Test set for initial value problem solvers*, CWI Report MAS–R9832, ISSN 1386–3703, Amsterdam, (1998).

[7] A. Logg, *Multi-adaptive galerkin methods for odes i*, submitted to SIAM J. Sci. Comput., (2001).

[8] ——, *Multi-adaptive galerkin methods for odes ii: Implementation & applications*, submitted to SIAM J. Sci. Comput., (2001).

[9] ——, *Multi-adaptive galerkin methods for odes iii: A priori error analysis*, in preparation, (2002).

[10] ——, *Multi-adaptive galerkin methods for odes iv: Stiff problems*, in preparation, (2002).

[11] G. Söderlind, *The automatic control of numerical integration*, CWI Quarterly, 11 (1998), pp. 55–74.

[12] J. Verwer, *Explicit runge-kutta methods for parabolic partial differential equations*, Appl. Numer. Math., 22 (1996), pp. 359–379.

Chalmers Finite Element Center Preprints

2001–01 *A simple nonconforming bilinear element for the elasticity problem*
Peter Hansbo and Mats G. Larson

2001–02 *The $\mathcal{LL}^*$ finite element method and multigrid for the magnetostatic problem*
Rickard Bergström, Mats G. Larson, and Klas Samuelsson

2001–03 *The Fokker-Planck operator as an asymptotic limit in anisotropic media*
Mohammad Asadzadeh

2001–04 *A posteriori error estimation of functionals in elliptic problems: experiments*
Mats G. Larson and A. Jonas Niklasson

2001–05 *A note on energy conservation for Hamiltonian systems using continuous time finite elements*
Peter Hansbo

2001–06 *Stationary level set method for modelling sharp interfaces in groundwater flow*
Nahidh Sharif and Nils-Erik Wiberg

2001–07 *Integration methods for the calculation of the magnetostatic field due to coils*
Marzia Fontana

2001–08 *Adaptive finite element computation of 3D magnetostatic problems in potential formulation*
Marzia Fontana

2001–09 *Multi-adaptive galerkin methods for ODEs I: theory & algorithms*
Anders Logg

2001–10 *Multi-adaptive galerkin methods for ODEs II: applications*
Anders Logg

2001–11 *Energy norm a posteriori error estimation for discontinuous Galerkin methods*
Roland Becker, Peter Hansbo, and Mats G. Larson

2001–12 *Analysis of a family of discontinuous Galerkin methods for elliptic problems: the one dimensional case*
Mats G. Larson and A. Jonas Niklasson

2001–13 *Analysis of a nonsymmetric discontinuous Galerkin method for elliptic problems: stability and energy error estimates*
Mats G. Larson and A. Jonas Niklasson

2001–14 *A hybrid method for the wave equation*
Larisa Beilina, Klas Samuelsson and Krister Åhlander

2001–15 *A finite element method for domain decomposition with non-matching grids*
Roland Becker, Peter Hansbo and Rolf Stenberg

2001–16 *Application of stable FEM-FDTD hybrid to scattering problems*
Thomas Rylander and Anders Bondeson

2001–17 *Eddy current computations using adaptive grids and edge elements*
Y. Q. Liu, A. Bondeson, R. Bergström, C. Johnson, M. G. Larson, and K. Samuelsson

2001–18 *Adaptive finite element methods for incompressible fluid flow*
Johan Hoffman and Claes Johnson

2001–19 *Dynamic subgrid modeling for time dependent convection–diffusion–reaction equations with fractal solutions*
Johan Hoffman

2001–20 *Topics in adaptive computational methods for differential equations*
Claes Johnson, Johan Hoffman and Anders Logg

2001–21 *An unfitted finite element method for elliptic interface problems*
Anita Hansbo and Peter Hansbo

2001–22 *A P^2–continuous, P^1–discontinuous finite element method for the Mindlin-Reissner plate model*
Peter Hansbo and Mats G. Larson

2002–01 *Approximation of time derivatives for parabolic equations in Banach space: constant time steps*
Yubin Yan

2002–02 *Approximation of time derivatives for parabolic equations in Banach space: variable time steps*
Yubin Yan

2002–03 *Stability of explicit-implicit hybrid time-stepping schemes for Maxwell's equations*
Thomas Rylander and Anders Bondeson

2002–04 *A computational study of transition to turbulence in shear flow*
Johan Hoffman and Claes Johnson

2002–05 *Adaptive hybrid FEM/FDM methods for inverse scattering problems*
Larisa Beilina

2002–06 *DOLFIN - Dynamic Object oriented Library for FINite element computation*
Johan Hoffman and Anders Logg

2002–07 *Explicit time-stepping for stiff ODEs*
Kenneth Eriksson, Claes Johnson and Anders Logg

These preprints can be obtained from

`www.phi.chalmers.se/preprints`