57
Adaptive solvers of initial value prolems

> On two occasions I have been asked (by members of Parliament), "Pray, Mr Babbage, if you put into the machine wrong figures, will the right answer come out?". I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question. (Babbage (1792-1871))

57.1 Introduction

In this chapter, we discuss the important issue of *adaptive error control* for numerical methods for initial value problems. This is the subject of automated choice of the time step with the purpose of controlling the numerical error to within a given tolerance level. The basic idea is to combine *feed-back* information from the computation concerning the *residual* of the computed solution and the results of auxiliary computations of *stability factors*. We focus first on on the cG(1) method and then comment on the backward Euler method, also referred to as dG(0), the discontinuous Galerkin method with piecewise constants.

We also discuss the application of cG(1) and dG(0) to a class of so-called *stiff* IVPs typically arising in chemical reaction modeling.

57.2 The cG(1) method

We recall that cG(1), the continuous Galerkin method with polynomials of order 1, for the initial value problem $\dot{u}(t) = f(u(t))$ for $t > 0$, $u(0) = u^0$, with $f : \mathbb{R}^d \to \mathbb{R}^d$, takes the form

$$U(t_n) = U(t_{n-1}) + \int_{t_{n-1}}^{t_n} f(U(t))\, dt, \quad n = 1, 2, ..., \tag{57.1}$$

where $U(t)$ is continuous piecewise linear with nodal values $U(t_n) \in \mathbb{R}^d$ at an increasing sequence of discrete time levels $0 = t_0 < t_1 < ...$, and $U(0) = u^0$. If we evaluate the integral in (57.1) with the midpoint quadrature rule, we obtain the Midpoint method:

$$U(t_n) = U(t_{n-1}) + k_n f\left(\frac{U(t_n) + U(t_{n-1})}{2}\right), \quad n = 1, 2, ..., \tag{57.2}$$

where $k_n = t_n - t_{n-1}$ is the time step. The cG(1)-method is the first in a family of cG(q)-methods with $q = 1, 2, ...,$, where the solution is approximated by continuous piecewise polynomials of order q. The Galerkin "orthogonality" of cG(1) is expressed by the fact that the method can be formulated

$$\int_{t_{n-1}}^{t_n} (\dot{U}(t) - f(U(t))) \cdot v\, dt = 0, \quad n = 1, 2, ..., \tag{57.3}$$

for all $v \in \mathbb{R}^d$. This says that the *residual*

$$R(U(t) = \dot{U}(t) - f(U(t)), \quad t \in [0, T], \tag{57.4}$$

of the continuous piecewise linear approximate solution $U(t)$ is *orthogonal* to the constant functions $v(t) = v \in \mathbb{R}^d$ on each subinterval (t_{n-1}, t_n). The residual $\dot{u}(t) - f(u(t))$ of the exact solution is zero since $\dot{u}(t) = f(u(t))$, while the residual of $R(U(t))$ of the approximate solution $U(t)$ is non-zero in general. Similarly, in cG(q) the residual is orthogonal on (t_{n-1}, t_n) to polynomials of degree $q-1$. Note that (57.1) is a vector equation that reads

$$U_i(t_n) = U_i(t_{n-1}) + \int_{t_{n-1}}^{t_n} f_i(U(t))\, dt, \quad n = 1, 2, ..., i = 1, ..., d,$$

as can be seen from (57.3) upon setting $v = e_i$, $i = 1, ..., d$.

We will now study the problem of *automatic step-size control* with the purpose of keeping the error

$$\|u(T) - U(T)\| \leq TOL,$$

where $T = t_N$ is a final time and TOL is a given tolerance, while using as few time steps as possible. The objective is the same as that of computing an integral over an interval $[0, T]$ using numerical quadrature to a certain tolerance using as few quadrature points as possible. This is exactly the problem we meet in the case of a scalar initial value problem $\dot{u}(t) = f(u(t), t)$ with $f(u(t), t) = f(t)$.

We shall derive an *a posteriori error estimate* in which the final error $\|u(T)-U(T)\|$ is estimated in terms of the residual $R(U(t) = \dot{U}(t)-f(U(t))$ and certain *stability factors* that measure the *accumulation* of the numerical errors introduced in each time step.

The a posteriori error estimate takes the form

$$\|u(T) - U(T)\| \leq S_c(T) \max_{0\leq t\leq T} \|k(t)R(U(t)\|, \tag{57.5}$$

where $k(t) = k_n = t_n - t_{n-1}$ for $t \in [t_{n-1}, t_n)$ and where the stability factor $S_c(T)$ is defined as follows. We consider the linearized problem

$$-\dot{\varphi}(t) = A^\top(t)\varphi(t) \quad \text{for } 0 < t < T, \, \varphi(T) = \varphi^0, \tag{57.6}$$

where

$$A(t) = \int_0^1 f'(su(t) + (1-s)U(t))\, ds.$$

We note that replacing $u(t)$ by $U(t)$ gives the following approximate formula for $A(t)$,

$$A(t) \approx f'(U(t)),$$

assuming $U(t)$ is close to $u(t)$. We conclude that $A(t)$ is close to the Jacobian $f'(u(t))$ of $f(v)$ at $v = u(t)$ if $U(t)$ is a reasonable approximation of $u(t)$ Note that the dual $A^\top(t)$ of $A(t)$ occurs in (57.6). Note further that the linearized dual problem (57.6) runs *backward* in time since the initial value $\varphi(T) = \varphi^0$ is specified at time $t = T$. We are now ready to introduce the following stability factors:

$$\begin{aligned} S_d(T) &= \max_{\varphi^0 \in \mathbb{R}^d} \frac{\|\varphi(t)\|}{\|\varphi^0\|}, \\ S_c(T) &= \max_{\varphi^0 \in \mathbb{R}^d} \frac{\int_0^t \|\dot{\varphi}(s)\|\, ds}{\|\varphi^0\|}, \end{aligned} \tag{57.7}$$

where φ solves (57.6). We note that the stability factors measure different features of the the dual solution φ. The stability factor $S_d(t)$ measures the maximal perturbation growth over the time interval $[0, T]$. We met this factor in the previous chapter. We shall see that this factor is tailored to measure the effect of an error in the initial data u^0 and the "d" in S_d refers to "data". The stability factor $S_c(t)$ measures the integral of $\|\dot{\varphi}\|$ over $[0, T]$ and is geared to evaluate the error in cG(1) and the "c" in S_c refers to "computation".

We shall give the proof of (57.5) below, first in a very simple case with $n = 1$ and $f(u(t)) = au(t)$ with a a constant and then in the general case.

The proofs are very similar. Before plunging into the proofs, we shall try to digest the a posteriori error estimate, and see how it can be used to design an adaptive algorithm aiming at controlling the final error $\|u(T) - U(T)\|$ on a given tolerance level with as few time steps as possible.

The stability factors $S_c(T)$ and $S_d(T)$ can be computed by numerically solving the linearized dual problem (57.6) with $\varphi^0 = e_i$ for $i = 1, ..., d$. If d is large, then we may reduce the variation of the initial data by limiting the error control to certain components only, or by trying to choose φ^0 parallel to $u(T) - U(T)$, which we approximate as $U_h(T) - U_H(T)$ with $U_h(T)$ and $U_H(T)$ being approximations computed with two different tolerances.

57.3 Adaptive time step control for cG(1)

We recall the basic error estimate (57.5):

$$\|u(T) - U(T)\| \leq S_c(T) \max_{0 \leq t \leq T} \|k(t))R(t)\|, \tag{57.8}$$

where $R(t) = \dot{U}(t) - f(U(t))$ and we assume that the stability factor $S(T)$ has been computed or estimated. We will return to this issue below. To achieve $\|u(T) - U(T)\| \leq TOL$, we use (57.5) to choose the time steps $k_n = t_n - t_{n-1}$ so that

$$k(t) = k_n \approx \frac{TOL}{S_c(T) R_n} \quad \text{for } t \in [t_{n-1}, t_n), \tag{57.9}$$

where

$$R_n = \max_{t_{n-1} \leq t \leq t_n} \|\dot{U}(t) - f(U(t))\|$$

is the residual on the time interval $[t_{n-1}, t_n)$. Note that the residual R_n is computable from the computed solution $U(t)$ and if $S_c(T)$ is known, timestepalg gives an equation for the time step $k_n = t_n - t_{n-1}$, where t_{n-1} already known. As with adaptive numerical quadrature, (57.9) yields a nonlinear equation for the time step $k_n = t_n - t_{n-1}$ that we can seek to solve using some form of trial-and-error strategy or by prediction, e.g. replacing R_n by R_{n-1}.

57.4 A posteriori error estimate for cG(1) a linear scalar IVP

We shall now prove an a posteriori error estimate for cG(1) for a a linear scalar IVP of the form

$$\dot{u}(t) = au(t) + f(t) \quad \text{for } t > 0, u(0) = u^0, \tag{57.10}$$

where a is a constant and $f(t)$ is a given function. The analysis is based on representing the error in terms of the solution $\varphi(t)$ of the following dual problem:

$$\begin{cases} -\dot{\varphi} = a\varphi & \text{for } T > t \geq 0, \\ \varphi(T) = e(T), \end{cases} \tag{57.11}$$

where $e = u - U$. Note again that (57.11) runs "backwards" in time starting at time t_N and that the time derivative term $\dot{\varphi}$ has a minus sign. We start from the identity

$$|e(T)|^2 = |e(T)|^2 + \int_0^T e\,(-\dot{\varphi} - a\varphi)\,dt,$$

and integrate by parts to get the following representation of $|e(T)|^2$,

$$|e(T)|^2 = \int_0^T (\dot{e} - ae)\varphi\,dt + e(0)\varphi(0),$$

where we allow $U(0)$ to be different from $u(0)$, corresponding to an error in the initial value $u(0)$. Since u solves the differential equation (57.10), that is $\dot{u} + au = f$, we have

$$\dot{e} - au = \dot{u} - au - \dot{U} + aU = f - \dot{U} + aU,$$

and thus we obtain the following representation of the error $|e(T)|^2$ in terms of the residual $R(U) = \dot{U} - aU - f$ and the dual solution φ ,

$$|e(T)|^2 = \int_0^T (f + aU - \dot{U})\varphi\,dt + e(0)\varphi(0) = -\int_0^{t_N} R(U)\varphi\,dt + e(0)\varphi(0). \tag{57.12}$$

Next, we use the Galerkin orthogonality of cG(1),

$$\int_{t_{n-1}}^{t_n} R(U)\,dt = 0 \quad \text{for } n = 1, 2, ...,$$

to rewrite (57.12) as

$$e(T)^2 = -\int_0^T R(U)(\varphi - \bar{\varphi})\,dt + e(0)\varphi(0), \tag{57.13}$$

where $\bar{\varphi}$ is the mean-value of φ over each time interval, that is

$$\bar{\varphi}(t) = \frac{1}{k_n}\int_{t_{n-1}}^{t_n} \varphi(s)\,ds \quad \text{for } t \in [t_{n-1}, t_n).$$

We shall now use

$$\int_{I_n} |\varphi - \bar{\varphi}|\,dt \leq k_n \int_{I_n} |\dot{\varphi}|\,dt,$$

which follows by integration from the facts that

$$\varphi(t)-\bar\varphi(t)=\frac{1}{k_n}\int_{t_{n-1}}^{t_n}(\varphi(t)-\varphi(s))\,ds,$$

and

$$|\varphi(t)-\varphi(s)|\le\int_s^t|\dot\varphi(\sigma)|\,d\sigma\le\int_{t_{n-1}}^{t_n}|\dot\varphi(\sigma)|\,d\sigma\quad\text{for } s,t\in[t_{n-1},t_n].$$

Thus, (57.13) implies

$$\begin{aligned}|e(T)|^2&\le\sum_{n=1}^N R_n\int_{I_n}|\varphi-\bar\varphi|dt+|e(0)||\varphi(0)|\\&\le\sum_{n=1}^N k_n\,R_n\int_{I_n}|\dot\varphi|dt+|e(0)||\varphi(0)|,\end{aligned}\tag{57.14}$$

where

$$R_n=\max_{t_{n-1}\le t\le t_n}|R(U(t))|.$$

Bringing out the max of k_nR_n over n, we get

$$|e(T)|^2\le\max_{1\le n\le N}k_nR_n\int_0^{t_N}|\dot\varphi|\,dt+|e(0)||\varphi(0)|.$$

Recalling that $\varphi(T)=e(T)$ and using the definitions of $S_c(t_N)$ and $S_d(t_N)$, we get the following final estimate,

$$|e(T)|\le S_c(T)\max_{0\le t\le T}|k(t)R(U(t))|+S_d(T)|e(0)|.$$

The stability factors $S_c(T)$ and $S_d(T)$ measure the effects of the accumulation of error in the approximation. To give the analysis a quantitative meaning, we have to give a quantitative bound of this factor. The following lemma gives an estimate for $S_c(T)$ and $S_d(T)$ in the cases $a\le 0$ and the case $a\ge 0$ with possibly vastly different stability factors. We notice that the solution $\varphi(t)$ of (57.11) is given by the explicit formula

$$\varphi(t)=e(T)\exp(a(T-t)).$$

We see that if $a\le 0$, then the solution $\varphi(t)$ decays as t decreases from T, and the case $a\le 0$ is thus the "stable case". If $a>0$ then the exponential factor $\exp(aT)$ enters, and depending on the size of a this case is "unstable". More precisely, we conclude directly from the explicit solution formula that

Lemma 57.1 *The stability factors $S_c(T)$ and $S_d(T)$ satisfy if $a>0$,*

$$S_d(T)\le\exp(aT),\quad S_c(T)\le\exp(aT),\tag{57.15}$$

and if $a\le 0$, then

$$S_d(T)\le 1,\quad S_c(T)\le 1.\tag{57.16}$$

57.5 A posteriori error estimate for cG(1) for a general IVP

The extension of the a posteriori error analysis to a general IVP $\dot{u} = f(u)$ with $f : \mathbb{R}^d \to \mathbb{R}^d$ goes as follows. We recall that the linearized dual problem takes the form

$$-\dot{\varphi}(t) = A^\top(t)\varphi(t) \quad \text{for } 0 < t < T,\, \varphi(T) = e(T), \tag{57.17}$$

with

$$A(t) = \int_0^1 f'(su(t) + (1-s)U(t))\, ds,$$

where $u(t)$ is the exact solution and $U(t)$ the approximate solution. We now use the fact that

$$\begin{aligned} A(t)e(t) &= \int_0^1 f'(su(t) + (1-s)U(t))e(t)\, ds \\ &= \int_0^1 \frac{d}{ds} f(su(t) + (1-s)U(t))\, ds = f(u(t)) - f(U(t)), \end{aligned} \tag{57.18}$$

where we used the Chain rule and the Fundamental Theorem of Calculus. We start from the identity

$$\|e(T)\|^2 = \|e(T)\|^2 + \int_0^T e \cdot (-\dot{\varphi} - A^\top \varphi)\, dt,$$

and integrate by parts to get the error representation,

$$\|e(T)\|^2 = \int_0^T (\dot{e} - Ae) \cdot \varphi\, dt + e(0) \cdot \varphi(0),$$

where we allow $U(0)$ to be different from $u(0)$, corresponding to an error in the initial value $u(0)$. Since u solves the differential equation $\dot{u} - f(u) = 0$, (57.18) implies

$$\dot{e} - Ae = \dot{u} - f(u) - \dot{U} + f(U) = -\dot{U} + f(U),$$

and thus we obtain the following representation of the error $\|e(T)\|^2$ in terms of the residual $R(U) = \dot{U} - f(U)$ and the dual solution φ,

$$\|e(T)\|^2 = -\int_0^{t_N} R(U)\varphi\, dt + e(0)\varphi(0). \tag{57.19}$$

From this point, the proof proceeds just as in the scalar case considered above and we thus obtain the following a posteriori error estimate

$$\|e(T)\| \le S_c(T) \max_{0 \le t \le T} \|k(t)R(U(t))\| + S_d(T)\|e(0)\|,$$

which can be used a basis for adaptive time step control as described above. The stability factors $S_c(T)$ and $S_d(T)$ may be estimated by solving the dual problem with suitable initial data. The proof of the a posteriori error estimate shows that the stability factors may be defined by

$$\begin{aligned} S_d(T) &= \frac{\|\varphi(t)\|}{\|e(T)\|}, \\ S_c(T) &= \frac{\int_0^t \|\dot{\varphi}(s)\|\, ds}{\|e(T)\|}, \end{aligned} \tag{57.20}$$

where φ solves the linearized dual problem with initial data $\varphi(T) = e(T)$. As indicated, to compute the stability factors $S_d(T)$ and $S_c(T)$, we may solve the dual problem with some estimation of $e(T)$ obtained by solving the initial value problem with two tolerances and approximating $e(T)$ by the difference of the corresponding approximate solutions. Alternatively, choosing $\varphi(T) = e_i$, we obtain a posteriori error control for error component $e_i(T)$. If d is not large, we may this way control all components of the error, and if d is large, we may choose a couple different i at random.

The size of the stability factors indicate the degree of stability of the solution $u(t)$ being computed. If the stability factors are large, the residuals $R(U(t)$ and $e(0)$ have to be made correspondingly smaller by choosing smaller time steps and the computational problem is more demanding.

57.6 A posteriori error estimates for backward Euler for a general IVP

We now derive an a posteriori error estimate for the backward Euler method for the IVP (56.1):

$$U(t_n) = U(t_{n-1}) + k_n f(U(t_n)), \quad n = 1, 2, ..., N, \quad U(0) = u^0.$$

We associate a function $U(t)$ defined on $[0, T]$ to the function values $U(t_n)$, $n = 0, 1, ..., N$, as follows:

$$U(t) = U(t_n) \quad \text{for } t \in (t_{n-1}, t_n].$$

In other words, $U(t)$ is piecewise constant on $[0, T]$ and takes the value $U(t_n)$ on $(t_{n-1}, t_n]$, and thus takes a jump from the value $U(t_{n-1})$ to the value $U(t_n)$ at the time level t_{n-1}.

We can now write the backward Euler method as,

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

or equivalently

$$U(t_n) \cdot v = U(t_{n-1}) \cdot v + \int_{t_{n-1}}^{t_n} f(U(t)) \cdot v \, dt, \tag{57.21}$$

for all $v \in \mathbb{R}^d$. This method os also referred to as dG(0), that is the *discontinuous Galerkin method of order zero, corresponding to approximating the exact solution by a piecewise constant function* $U(t)$ *satisfying the orthogonality condition* (57.21).

We are now ready to derive an a posteriori error estimate following the same strategy as for the cG(1) method. We start from the identity

$$\|e(T)\|^2 = \|e(T)\|^2 + \sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} e \cdot (-\dot{\varphi} - A^\top \varphi) \, dt,$$

and integrate by parts on each subinterval (t_{n-1}, t_n) to get the following error representation,

$$\begin{aligned} \|e(T)\|^2 = &\sum_{n=1}^{N} \int_{t_{n-1}}^{t_n} (\dot{e} - Ae) \cdot \varphi \, dt \\ &- \sum_{n=2}^{N-1} (U(t_n) - U(t_{n-1}))\varphi(t_{n-1}), \end{aligned}$$

where the last term results from the jumps of $U(t)$ at the the nodes $t = t_{n-1}$ and we assume $U(0) = u(0)$ for simplicity. Since u solves the differential equation $\dot{u} - f(u) = 0$, (57.18) and the fact that $\dot{U}$ on (t_{n-1}, t_n) imply

$$\dot{e} - Ae = \dot{u} - f(u) - \dot{U} + f(U) = -\dot{U} + f(U) = f(U) \quad \text{on } (t_{n-1}, t_n),$$

and thus we obtain

$$\|e(T)\|^2 = -\sum_{n=2}^{N-1} (U(t_n) - U(t_{n-1}))\varphi(t_{n-1}) + \int_0^{t_N} f(U)\varphi \, dt.$$

Using (57.21) with $v = \bar{\varphi}$, the mean value of φ as above, we get

$$\begin{aligned} \|e(T)\|^2 = -\sum_{n=2}^{N-1} (U(t_n) - U(t_{n-1})) \cdot (\varphi(t_{n-1}) - \bar{\varphi}(t_{n-1})) \\ + \sum_{n=1}^{n} \int_{t_{n-1}}^{t_n} f(U)(\varphi - \bar{\varphi}) \, dt. \end{aligned}$$

We note that

$$\int_{t_{n-1}}^{t_n} f(U)(\varphi - \bar{\varphi}) \, dt = 0,$$

since $f(U(t))$ is constant on $(t_{n-1}, t_n]$, and $\bar\varphi$ is the mean value of φ, and thus the error representation takes the final form

$$\|e(T)\|^2 = -\sum_{n=2}^{N-1}(U(t_n) - U(t_{n-1}))\cdot(\varphi(t_{n-1}) - \bar\varphi(t_{n-1})).$$

Using

$$\|\varphi(t_{n-1}) - \bar\varphi(t_{n-1})\| \le \int_{t_{n-1}}^{t_n}\|\dot\varphi(t)\|\,dt,$$

we obtain the following a posteriori error estimate for the backward Euler method ,

$$\|e(T)\| \le S_c(T)\max_{1\le n\le N}\|U(t_n)) - U(t_{n-1}))\|. \tag{57.22}$$

Note the very simple form of this estimate involving the jumps $\|U(t_n)) - U(t_{n-1}))\|$ playing the role the residual. The a posteriori error estimate (57.22) can be used as a basis for an algorithm for adaptive time step control of the following form: for $n = 1, 2, ...$, choose k_n so that

$$\|U(t_n)) - U(t_{n-1}))\| \approx \frac{TOL}{S_c(T)}.$$

57.7 Stiff initial value problems

A *stiff* initial value problem $\dot u = f(u)$ may be characterized by the fact that the stability factors $S_d(T)$ and $S_c(T)$ are of moderate size even for large T, while the norm of the linearized operator $f'(u(t))$ is large, that is the Lipschitz constant L_f is very large. Such initial value problems are common for example in models of chemical reaction with reactions on a range of time scales from slow to fast. Typical solutions include so-called *transients* where the fast reactions make the solution change quickly over a short (initial) time interval, after which the fast reactions are "burned out" and the slow reactions make the solution change on a longer time scale.

The prototype of a stiff initial value problem has the form

$$\dot u = f(u) \equiv -Au \quad \text{for } t > 0,\ u(t) = u^0 = (u_i^0), \tag{57.23}$$

where A is a constant symmetric positive semidefinite $d\times d$ matrix with non-negative eigenvalues λ_i ranging from zero to large positive values. Accordingly, the norm of the matrix A is large and L_f is large. By diagonalization, we may reduce to the case when A is a diagonal matrix with non-negative diagonal elements λ_i, in which case the solution $u(t) = (u_i(t))$ is given by

$$u_i(t) = \exp(-\lambda_i\, t)u_i^0 \quad \text{for } t > 0, \tag{57.24}$$

with $u^0 = (u_i^0)$. This explicit solution formula shows that a component $u_i(t)$ corresponding to a large positive eigenvalue λ_i decays very quickly to zero, while a component with a small eigenvalue stays almost constant for a long time and eventually decays to zero. The sign of the eigenvalues is evidently crucial: if some λ_i was negative, then the corresponding solution component would explode exponentially more or less quickly depending on the size of λ_i. In particular, (57.24) with the λ_i non-negative implies

$$\|u(t)\| \leq \|u^0\| \quad \text{for } t > 0, \tag{57.25}$$

which indicates a form of stability with stability factor equal to 1 in the sense that the norm of the solution does not increase in time.

The dual problem corresponding to (57.23) takes the form

$$-\dot{\varphi} + A\varphi = 0 \quad \text{for } T > t > 0,\ \varphi(T) = \psi,$$

with ψ given data at time $t = T$. As a counterpart of (57.25), we conclude that $S_d(T) \leq 1$. We can similarly show that $S_c(T)$ grows very slowly with increasing T. We sum up: (57.23) represents a stiff problem; stability factors are of moderate size even for large T while the norm of the (linearized) operator A is large.

From numerical point of view, stiff problems may seem particularly friendly since the stability factors grow very slowly with time, but there is one hook that has attracted a lot of attention in the literature on numerical methods for initial value problems, namely the failure of an explicit method like the forward Euler method. We write this method for the equation $\dot{u} = -Au$ in the form

$$U^n = U^{n-1} - k_n A U^{n-1}$$

with U^n an approximation of $u(t_n)$ and $0 = t_0 < t_1 < ...$ an increasing sequence of time levels, and $k_n = t_n - t_{n-1}$. If A is diagonal with diagonal elements $\lambda_i \geq 0$, then

$$U_i^n = (1 - k_n\lambda_i)U_i^{n-1}$$

and if λ_i is large positive, then $|1 - k_n\lambda_i|$ may be much larger than 1 unless the time step k_n is sufficiently small ($k_n \leq 2/|\lambda_i|$ for all i) and the the numerical solution will then quickly explode to infinity, while the corresponding exact solution quickly decays to zero. The explicit Euler method will thus give completely wrong results unless sufficiently small time steps are used. This may lead to very inefficient time-stepping since after the transients have died out, the solution may vary only slowly and large time steps would be desirable. We note that the time step limit $k_n \leq 2/|\lambda_i|$ for all i, is set by the largest eigenvalue $\max \lambda_i$, while the time long-time scale is set by the smallest eigenvalue $\min \lambda_i$, so that if the quotient $\max \lambda_i / \min \lambda_i$ is large (which signifies a stiff problem), then explicit Euler would be inefficient outside transients.

On the other hand, the dG(0), or implicit Euler method,

$$U^n + k_n A U^n = U^{n-1}$$

with

$$U_i^n = (1 + k_n\lambda_i)^{-1} U_i^{n-1}$$

will be stable and work very well without step size limitation because $1 + k_n\lambda_i \geq 1$ for all $\lambda_i \leq 0$.

For the cG(1)-method, we will have

$$U_i^n = \frac{1 - k_n\lambda_i}{1 + k_n\lambda_i} U_i^{n-1}$$

and stability prevails because

$$|\frac{1 - k_n\lambda_i}{1 + k_n\lambda_i}| \leq 1$$

for all $\lambda_i \geq 0$.

We conclude that both dG(0) and cG(1) may be used for stiff problems, but both these methods are implicit and require the solution of system of equations at each time step. More precisely, dG(0) for a problem of the form $\dot{u} = f(u)$ takes the form

$$U^n - k_n f(U^n) = U^{n-1}.$$

At each time step we have to solve an equation of the form $v - k_n f(v) = U^{n-1}$ with U^{n-1} given. To this end we may try a damped fixed point iteration in the form

$$v^{(m)} = v^{(m-1)} - \alpha(v^{(m-1)} - k_n f(v^{(m-1)}) - U^{n-1}),$$

with α some suitable matrix (or constant in the simplest case). Choosing $\alpha = I$, and iterating once with $v^0 = 0$ corresponds to the explicit Euler method. Convergence of the fixed point iteration requires that

$$\|I + k_n \alpha f'(v)\| < 1$$

for relevant values of v, which could force α to be small (e.g. in the stiff case with $f'(v)$ having large negative eigenvalues) and result in slow convergence. A first try could be to choose α to be a diagonal matrix with $\alpha_i = (f'_{ii})(v^{m-1}))^{-1}$ (corresponding to *diagonal scaling*) and hope that the number of iterations would not be too large. In some cases more efficient iterative solvers would have to be used.

57.8 On explicit time-stepping for stiff problems

We just learned that explicit time-stepping for stiff problems require small time steps outside transients and thus may be inefficient. We shall now indicate a way to get around this limitation through a process of stabilization, where a large time step is accompanied by a couple of small time steps. The resulting method has similarities with the control system of a modern (unstable) jet fighter like the Swedish JAS Gripen, the flight of which is controlled by quick small flaps of a pair of small extra wings ahead of the main wings, or balancing a stick vertically on the finger tips if we want a more domestic application.

We shall now explain the basic (simple) idea of the stabilization and present some examples, as illustrations of fundamental aspects of adaptive IVP-solvers and stiff problems. Thus to start with, suppose we apply explicit Euler to the scalar problem

$$\begin{aligned} \dot{u}(t) + \lambda u(t) = 0 \quad \text{for } t > 0. \\ u(0) = u^0, \end{aligned} \tag{57.26}$$

with $\lambda > 0$ taking first a large time step K satisfying $K\lambda > 2$ and then m small time steps k satisfying $k\lambda < 2$, to get the method

$$U^n = (1 - k\lambda)^m (1 - K\lambda) U^{n-1}, \tag{57.27}$$

altogether corresponding to a time step of size $k_n = K + mk$. Here K gives a large unstable time step with $|1-K\lambda| > 1$ and k is a small time step with $|1 - k\lambda| < 1$. Defining the polynomial function $p(x) = (1 - \theta x)^m (1 - x)$, where $\theta = \frac{k}{K}$, we can write the method (57.27) in the form

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

For stability we need

$$|p(K\lambda)| \leq 1, \quad \text{that is } |1 - k\lambda|^m (K\lambda - 1) \leq 1,$$

or

$$m \geq \frac{\log(K\lambda - 1)}{-\log|1 - k\lambda|} \approx 2\log(K\lambda), \tag{57.28}$$

with $c = k\lambda \approx 1/2$ for definiteness.

We conclude that m may be quite small even if $K\lambda$ is large, since the logarithm grows so slowly, and then only a small fraction of the total time would be spent on stabilizing time-stepping with the small time steps k.

To measure the efficiency gain we introduce

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

which is the number of time steps per unit interval with stabilized explicit Euler method, and by (57.28)) we have

$$\alpha \approx \frac{1 + 2\log(K\lambda)}{K + \log(K\lambda)/\lambda} \approx 2\lambda \frac{\log(K\lambda)}{K\lambda} \ll 2\lambda, \tag{57.29}$$

for $K\lambda \gg 1$. On the other hand, the number of time steps per unit interval for the usual explicit Euler is

$$\alpha_0 = 1/k = \lambda/2, \tag{57.30}$$

choosing a maximum time step $k = 2/\lambda$.

The cost reduction factor using the stabilized explicit Euler method would thus be

$$\frac{\alpha}{\alpha_0} \approx \frac{4\log(K\lambda)}{K\lambda}$$

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

We now present some examples using an adaptive cg(1) IVP-solver in stabilized explicit form with just a few iterations in each time step, which allows large time steps. In all problems we note the initial transient, where the solution components change quickly, and the oscillating nature of the time step sequence outside the transient with large time steps followed by some small stabilizing time steps.

EXAMPLE 57.1. We apply the indicated method to the scalar problem equation (57.26) with $u^0 = 1$ and $\lambda = 1000$, and display the result in Figure 57.1. The cost reduction factor with comparison to a standard explicit method is large: $\alpha/\alpha_0 \approx 1/310$.

EXAMPLE 57.2. We now consider the 2×2 diagonal system

$$\begin{aligned} \dot{u}(t) + \begin{pmatrix} 100 & 0 \\ 0 & 1000 \end{pmatrix} u(t) &= 0 \quad \text{for } t > 0, \\ u(0) &= u^0, \end{aligned} \tag{57.31}$$

with $u^0 = (1, 1)$. There are now two eigenmodes with large eigenvalues that have to be stabilized. The cost reduction is $\alpha/\alpha_0 \approx 1/104$.

EXAMPLE 57.3. The is the so-called HIRES problem ("High Irradiance RESponse") from plant physiology whcih consists of the following eight

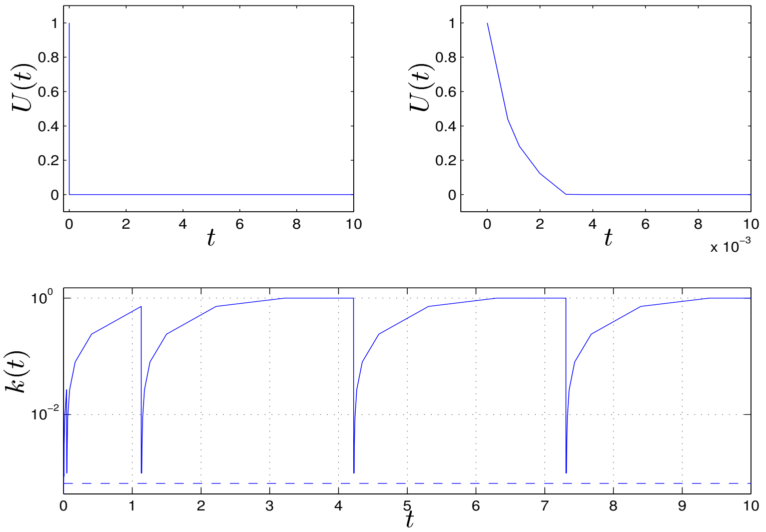

FIGURE 57.1. Solution and time step sequence for eq. (57.26), $\alpha/\alpha_0 \approx 1/310$.

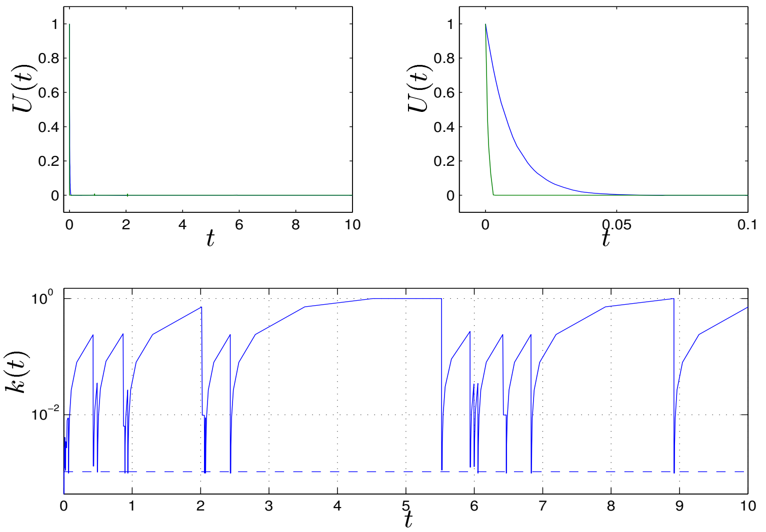

FIGURE 57.2. Solution and time step sequence for eq. (57.31), $\alpha/\alpha_0 \approx 1/104$.

equations:

$$\left\{\begin{array}{lcl} \dot{u}_1 & = & -1.71u_1 + 0.43u_2 + 8.32u_3 + 0.0007, \\ \dot{u}_2 & = & 1.71u_1 - 8.75u_2, \\ \dot{u}_3 & = & -10.03u_3 + 0.43u_4 + 0.035u_5, \\ \dot{u}_4 & = & 8.32u_2 + 1.71u_3 - 1.12u_4, \\ \dot{u}_5 & = & -1.745u_5 + 0.43u_6 + 0.43u_7, \\ \dot{u}_6 & = & -280.0u_6u_8 + 0.69u_4 + 1.71u_5 - 0.43u_6 + 0.69u_7, \\ \dot{u}_7 & = & 280.0u_6u_8 - 1.81u_7, \\ \dot{u}_8 & = & -280.0u_6u_8 + 1.81u_7, \end{array}\right. \tag{57.32}$$

together with the initial condition $u^0 = (1.0, 0, 0, 0, 0, 0, 0, 0.0057)$. We present the solution and the time step sequence in Figure 57.3. The cost is now $\alpha \approx 8$ and the cost reduction factor is $\alpha/\alpha_0 \approx 1/33$.

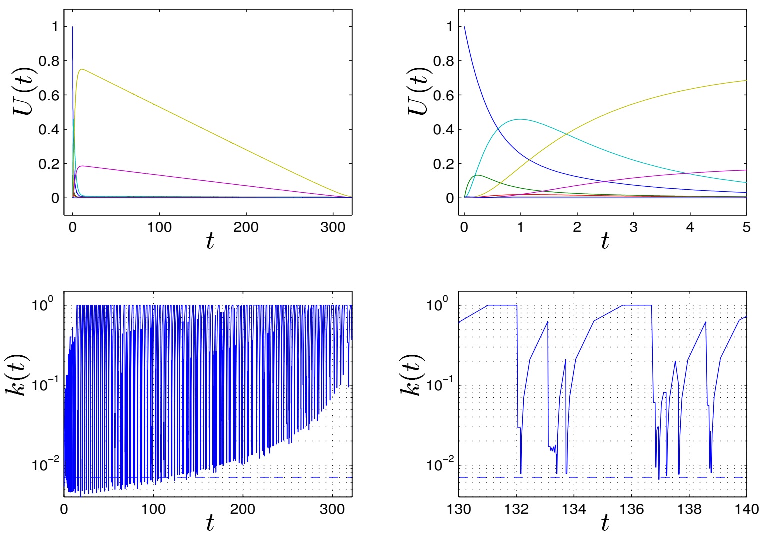

FIGURE 57.3. Solution and time step sequence for eq. (57.32), $\alpha/\alpha_0 \approx 1/33$.

EXAMPLE 57.4. The "Chemical Akzo-Nobel" problem consists of the following six equations:

$$\left\{\begin{array}{lcl} \dot{u}_1 & = & -2r_1 + r_2 - r_3 - r_4, \\ \dot{u}_2 & = & -0.5r_1 - r_4 - 0.5r_5 + F, \\ \dot{u}_3 & = & r_1 - r_2 + r_3, \\ \dot{u}_4 & = & -r_2 + r_3 - 2r_4, \\ \dot{u}_5 & = & r_2 - r_3 + r_5, \\ \dot{u}_6 & = & -r_5, \end{array}\right. \tag{57.33}$$

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).

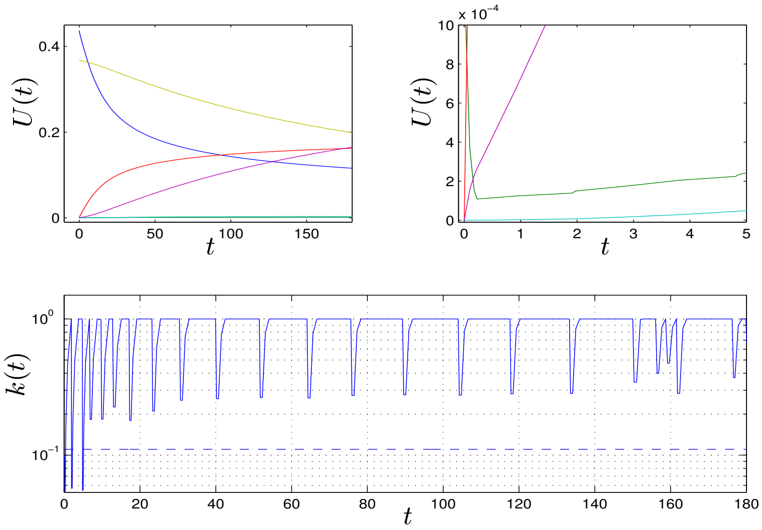

FIGURE 57.4. Solution and time step sequence for eq. (57.33), $\alpha/\alpha_0 \approx 1/9$.

EXAMPLE 57.5. We consider now Van der Pol's equation:

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

which we write as

$$\begin{cases} \dot{u}_1 &= u_2, \\ \dot{u}_2 &= -\mu(u_1^2 - 1)u_2 - u_1. \end{cases} \tag{57.34}$$

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 $\alpha/\alpha_0 \approx 1/75$.

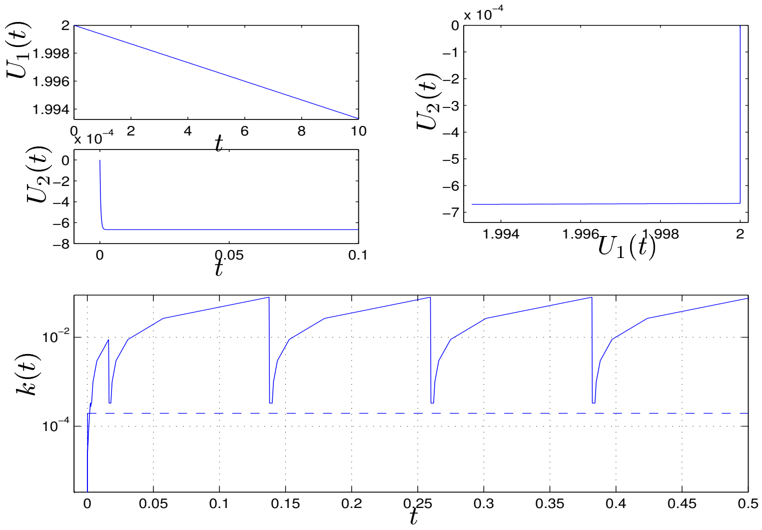

FIGURE 57.5. Solution and time step sequence for eq. (57.34), $\alpha/\alpha_0 \approx 1/75$.

Chapter 57 Problems

57.1. Compute the stability factors $S_d(T)$ and $S_c(T)$ for the linear scalar IVP $\dot{u}(t) = -\lambda(t)u(t)$ for $t > 0$, $u(0) = u^0$, where $\lambda(t)$ depends on time t and (a) $\lambda(t) \geq 0$, (b) $\lambda(t) < 0$.

57.2. Compute $S_d(T)$ and $S_c(T)$ for the linear 2×2 system $\dot{u}_1 = u_2, \quad \dot{u}_2 = -u_1$ for $t > 0$, $u(0) = u^0$.

57.3. Implement adaptive IVP-solvers based on dG(0) and cG(1) and apply the solvers to different problems.

57.4. Show that the a posteriori error estimate for cG(1) may be written on the form $\|e(T)\| \leq S_c(T)\max_{0\leq t\leq T}\|k(t)(f(U(t)) - \bar{f}(U(t)))\| + S_d(T)\|e(0)\|$, where $\bar{f}(U(t))$ is the mean-value of $f(U(t))$ over each time interval.

57.5. Show that choosing in the dual problem $\varphi(T) = e_i$ gives control of error component $e_i(T)$.

57.6. Develop explicit versions of dG(0) and cG(1) based on fixed point iteration at each time step. Show that with diagonal scaling such an explicit method may work very well for some stiff problems.