THE CONTROL OF ORDER AND STEPLENGTH FOR
BACKWARD DIFFERENTIATION METHODS

STIG SKELBOE

Abstract.

Backward differentiation methods are used extensively for integration of stiff systems of ordinary differential equations, but most implementations are inefficient when some of the eigenvalues of the Jacobi matrix are close to the imaginary axis. For these problems the performance of backward differentiation methods can be improved considerably by application of the instability test and reaction which is described in this paper. During instability the local truncation error oscillates rapidly with increasing magnitude. This property is used in the instability test. When instability is detected the order is lowered as much as possible without reducing the steplength.

The instability test and reaction is derived from a simplified analysis of integration of linear systems of differential equations, and the performance is verified for a number of linear test problems.

1. Introduction.

Let a system of \( N \) ordinary differential equations be defined by \( y' = f(y), \ y(x_0) = y_0, \ x \in [x_0, x_f] \). Let the class of solutions under consideration be limited to solutions where no eigenvalues of the Jacobi matrix \( J = \partial f / \partial y \) are situated in the right-hand complex half-plane. When the eigenvalues of the Jacobi matrix are widely separated the system is called stiff. The solution of a stiff system of differential equations contains rapidly and slowly varying components corresponding to the numerically large and small eigenvalues of \( J \) respectively. The steplength needed to keep the local truncation error of a discretization formula below a certain level depends heavily on the solution being dominated either by the rapidly or the slowly varying components. Long steps may be used during integration of the slow components if the discretization of the problem is stable. The ideal integration method should possess \( A \)-stability, but such methods are either of very low order, or they require the solution of a large system of nonlinear algebraic equa-

---

This work was supported by Statens Teknisk-Videnskabelige Forskningsråds under grant no. 518-5837. E-368.

tions in each step (e.g. order 2N or 3N) or several systems of order N. A large class of linear implicit multistep formulas are stiffly stable [8], and among these the backward differentiation formulas are used extensively in practice.

Implicit backward differentiation formulas of orders 1–6 are stiffly stable, and the formulas of order 1 and 2 are furthermore A-stable. The first attempt to use this class of formulas as the basis of a general program-system for solution of stiff systems of differential equations was done by Gear [6]. Since then backward differentiation formulas have been implemented in various ways and in numerous application programs.

In the implementation of formulas of orders 1-6 the computer program automatically selects order and steplengeth. When the Jacobi matrix contains complex eigenvalues $\lambda = \alpha \pm i\omega$, where $|\omega/\alpha| \gg 1$, this often leads to problems with control of order and steplengeth, as this type of eigenvalues causes instability in the higher order methods and results in reduction of the steplengeth.

In [4] and [5] a number of different methods for integration of stiff systems of differential equations are compared. The backward differentiation methods prove to be very efficient except for problems of the above-mentioned type. The trouble is that instability of the discretization in most implementations leads to a reduction of the steplengeth, whereas a reduction of the order is the most appropriate reaction in case of instability. In a new implementation of the backward differentiation methods the order is controlled both to allow maximum steplengeth and to assure stability. When instability is detected the order of the integration method is reduced as much as possible without decreasing the steplengeth.

Detection of instability is discussed in [9], but the results are not applied directly to the backward differentiation methods. Instability terms are recognized as being rapidly oscillating and as having increasing magnitudes. The presence of rapid oscillations implies that $|h^p y_n^{(0)}| > |h^q y_n^{(0)}|$, $p > q$ where $y_n$ is the numerical solution. The increasing magnitude is detected by the error estimate. It is essential that no information of the eigenvalues of the Jacobi matrix is required for the detection of instability.

The implementation with instability control was tested on a number of problems whose Jacobi matrix had eigenvalues of the type $\lambda = \alpha \pm i\omega$, $|\omega/\alpha| \gg 1$. The results are given in section 6, and they show that it is possible to control the order of the backward differentiation formulas to obtain efficiency in the solution of this type of problems.
2. Integration algorithm.

The implicit backward differentiation (BD) formula of order \( k \) is defined by:

\[ y_n = \sum_{r=1}^{k} \gamma_r y_{n-r} + h \beta_k f(y_n). \]  

The coefficients of the formulas of orders 1-6 are found in [7]. The integration formula is only one component of an integration algorithm which is made up of several parts:

A. predictor formulas
B. corrector formulas
C. implementation of predictor and corrector
D. error estimation
E. detection of instability
F. choice of order and steplength.

These six parts of the implementation of the backward differentiation formulas will be discussed in more detail.

A. Predictor formulas.

For the integration of very stiff systems the following predictor formula of order \( k \) should be used:

\[ y_n = \sum_{r=1}^{k+1} \gamma_{k+1} y_{n-r}. \]

The predictor formula \( y_n = \sum_{r=1}^{k} x^r y_{n-r} + h \beta_k f(y_{n-1}) \) may give troubles when \( hL \gg 1 \) where \( L \) is the Lipschitz constant. Let \( e_{n-1} \) be the global error of \( y_{n-1} \). When \( hL \gg 1 \) the error term \( ||h \beta_k[f(y_{n-1} - e_{n-1}) - f(y_{n-1})]|| \leq ||h \beta_k L e_{n-1}|| \) may be very serious and make the predictor useless [10].

B. Corrector formulas.

The implicit backward differentiation formulas defined in (1) are used as corrector formulas with order from 1 to 6. The corrected solution point \( y_n \) is found by solving a nonlinear system of algebraic equations. This is performed by a Newton or a pseudo-Newton method.

C. Implementation of predictor and corrector.

A very elegant implementation of the formulas (1) and (2) is based on divided differences ([1] and [11]). This formulation is equivalent to the application of (1) and (2) with the coefficients computed at non-equidistant points [3]. The predictor formula (2) is expressed by means of divided differences in the following way:

\[ y_n = y_{n-1} + \sum_{r=2}^{k+1} y_r(x_{n-1}, x_{n-2}, \ldots, x_{n-r}) \prod_{s=1}^{r-1} (x_n - x_{n-s}). \]
This formula is recognized as Newton’s interpolation formula with divided differences. The divided differences are computed component-wise for the vectors.

D. Error estimation.

The steplength control is based on the local truncation error as defined by

\[ L[y(x_n); h] = y(x_n) - \sum_{r=1}^{k+1} \frac{h^{k+r}}{r!} f^{(r)}(x_n) \]

\[ = C_{k+1} h^{k+1} y^{(k+1)}(x_n) + O(h^{k+2}) . \]

The principal local truncation error \( C_{k+1} h^{k+1} y^{(k+1)}(x_n) \) is supposed to dominate, and the error estimation is concentrated on this term. The derivative \( y^{(k+1)}(x_n) \) can be estimated by divided differences as

\[ y^{(r)}(x_n) \approx (r!) y[x_n, x_{n-1}, \ldots, x_{n-r}] . \]

The error of this estimate is \( C_{r+1} h^{r+1} y^{(r+1)}(x_n) + O(h^2) \). When the BD formula is of order \( k \) the divided differences \( y(x_n, x_{n-1}, \ldots, x_{n-k}) \) and \( y(x_{n-1}, x_{n-2}, \ldots, x_{n-k-1}) \) are used in the predictor and corrector formulas, and \( y(x_n, x_{n-1}, \ldots, x_{n-k-1}) \), which is necessary for estimation of \( y^{(k+1)}(x_n) \), is easily computed.

In the Nordsieck vector formulation \( f(y_n) \) takes part in the estimation of the derivatives. As explained for the predictor this may give a serious magnification of the error of \( y_n \). Therefore error estimation by means of differences in \( y_n, y_n-1, \ldots \) seems to be the most reliable approach.

E. Detection of instability.

Detection of instability is explained in all details in section 4.

F. Choice of order and steplength.

The ultimate objective of the control of order and steplength is to perform the integration as efficiently as possible keeping the error under a prescribed level. The corresponding optimal steplength and order distribution is usually not available as it depends on the entire solution, and therefore a local strategy is used.

If the discretization is stable the new order \( k \) is chosen to maximize the new stepsizes with the local truncation error at the prescribed value \( \varepsilon \):

\[ h_{new} = \frac{\varepsilon}{\left[ C_{k+1} y^{(k+1)}(x_n) \right]^{1/(k+1)}} . \]

For stability reasons the increase in steplength \( h_{new}/h_{old} \) is not allowed to exceed a value called the swing factor.
In the case of instability the order is reduced as much as possible without reducing the stepsize. This is further explained in section 4.

Some implementations of the backward differentiation formulas allow changes of order and stepsize after each step, and this may be advantageous when the nonlinearities of the differential equations are very strong. In this implementation the stepsize is piecewise constant. With a formula of order \( k \) the steplength is kept constant for \( k + 2 \) steps, provided the precision requirement is fulfilled. This affects the stability properties too as an increase in steplength causes a degradation of the stability properties.

The precision requirements can be formulated in different ways. For the integration of non-stiff systems a constant error per unit step seems appropriate, whereas a constant error per step is better suited for stiff systems. Depending on the application of the solution of the differential equation a relative or an absolute error measure is required. In DIFSUB [6] the relative and absolute error measure is combined in a way which is appropriate for most practical applications. A vector of weights \( \boldsymbol{w}_n \) is made up of the numerically greatest solution points so far encountered, and the resulting error measure is

\[
||C_{k+1}h^{k+1}y^{(k+1)}(x_n)/\boldsymbol{w}_n||
\]

where division by the vector is component-wise. This measure is used for the test examples of section 6.

3. Stability properties.

The absolute stability properties of the backward differentiation formulas are described by the characteristic equation (see [7]):

\[
z^k - \sum_{r=1}^{k} \alpha_r z^{k-r} = \frac{q(z) - \beta q(z)}{\lambda} = 0.
\]

In the complex \( \lambda \)-plane the boundary of absolute stability is determined by

\[
\lambda = \frac{q(\exp(i\theta))/\sigma(\exp(i\theta))}{-\pi < \theta \leq \pi}.
\]

The stability region is symmetrical about the real axis, and 0 is on the boundary. The stability regions for implicit backward differentiation formulas of orders 1-6 are given in [7].

The fully-drawn line in Fig. 1 is part of the stability boundary for the formula of order 3. The discretization is stable when \( \lambda \) is to the left of the boundary. The fine lines are curves of constant modulus \( \bar{d} \) and constant argument \( \theta \) for the greatest root \( z_1 \) of the characteristic polynomial (6).
For $\lambda = \alpha + i \omega$, where $|\omega/\alpha| > 14.5$ and $\alpha < 0$, $\lambda$ may enter the region of instability, and the corresponding values of $d$ and $\theta$ can be found from Fig. 1 ($z_1 = d \exp(i\theta)$).

When the elements of the Jacobi matrix are real-valued the complex eigenvalues occur as complex conjugate pairs, and the global error term corresponding to $z_1$ is of the type

$$e_{n,1} = d^n \sin(n\theta + \varphi).$$

When $\lambda$ is in the left-hand half-plane Fig. 1 shows that $\theta > 34^\circ$ for $d = 1.02$. The corresponding angles are listed in Table 1 for the implicit backward differentiation formulas of orders 3-6.

<table>
<thead>
<tr>
<th>order, $p$</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_{\text{min}}(d = 1.02)$</td>
<td>33.8$^\circ$</td>
<td>37.2$^\circ$</td>
<td>40.3$^\circ$</td>
<td>67.9$^\circ$</td>
</tr>
<tr>
<td>$\theta_{\text{min}}$ in radians, $\theta_p$</td>
<td>0.59</td>
<td>0.65</td>
<td>0.86</td>
<td>1.19</td>
</tr>
</tbody>
</table>
For all practical purposes it is necessary to change steplength during the integration, and this affects the stability of the method. The stability regions in [7] are only valid for constant steps, whereas absolute stability during change of steplength is studied in [2]. In general the stability properties are degraded by an increase of the steplength and improved by a decrease of the steplength.

The approach adopted here is integration with piecewise constant steplength. This means that the steplength is kept constant after a steppage, at least until all the \(x\)-points of the integration formula are equidistant. The stability during change of steplength is easier to analyse for this strategy than for a strategy allowing changes after each step. Analysis shows that the maximum degradation occurs in the second step after an increase of steplength. If the formulas are required to be \(A(0)\)-stable during the change of steplength the increase is limited by the values listed in Table 2.

<table>
<thead>
<tr>
<th>order, (p)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum increase in steplength, (\gamma_p)</td>
<td>2.6</td>
<td>1.9</td>
<td>1.5</td>
<td>1.2</td>
<td>1.042</td>
</tr>
</tbody>
</table>

The values from Table 2 are used as the maximum swing of the steplength \((\gamma = h_n/h_{n-1})\). This choice is somewhat arbitrary, but the Jacobi matrix for many systems of differential equations contains real eigenvalues, and therefore the values of Table 2 are reasonable as maximum swing of the steplength.

4. Instability.

When the discretization is unstable global error terms of the type \(e_{n,1} = d^n \sin(n\theta + \varphi)\) (7) are increasing in magnitude and getting significant. For a slowly increasing error \(e_{n,1}\) the dominant term of \(C_{p+1}h^{p+1}e_{n,1}^{(p+1)}\) is

\[
C_{p+1}d^{p+1}\sin(n\theta + \varphi_{p+1})
\]

when the order of the \(BD\) formula is \(p\).

For the backward differentiation formula of order 3 Fig. 1 shows that \(h\lambda\), where \(\lambda = \alpha + i\omega\), may enter the region of instability for \(|\omega/\alpha| > 14.5\) and \(\alpha < 0\). When the boundary of stability is crossed the value of \(\theta\) is smaller than \(\theta = 0.96\) rad. For a fixed value of \(|\omega/\alpha|\) the value of \(\theta\) increases when \(h\lambda\) proceeds into the region of instability. For formulas of orders 3 to 6 Table 1 shows the values of \(\theta\) for \(d = 1.02\) and \(\alpha = 0\).
Two properties of the error term (8) are very pronounced during instability. First the value of $\theta$ is of the order of magnitude 1. This implies that the error term is very rapidly oscillating, and that the magnitude of $\theta^{p+1}$ is almost independent of $p$. Second, the value of $d$ is greater than 1, and therefore the estimated error grows steadily when the instability is sufficiently serious. In the following it is explained how a test for smoothness can detect the rapid instability oscillations and how a test for growth can detect the increasing magnitudes of the instability terms.

Assume that the steplength $h$ and order $p$ are controlled such that

$$||C_{p+1}h^{p+1}y_n^{(p+1)}||_n ||y_n|| \leq \varepsilon$$

(9)

where $\varepsilon < 1$. The derivative $y_n^{(r)}$ means $y^{(r)}(x_n)$ estimated from the numerical solution $y_n, y_{n+1}, \ldots$ by means of the divided difference formula (5).

When condition (9) is fulfilled and the discretization is stable the numerical solution will usually be smooth in the following sense:

$$\ldots ||h^{p-1}y_n^{(p-1)}||_n ||h^p y_n^{(p)}||_n ||h^{p+1}y_n^{(p+1)}||_n$$

(10)

It is easy to construct an example with a rapidly and a slowly decaying solution component such that condition (9) is fulfilled and the discretization is stable but condition (10) violated. However, the situation will last only for few steps as the order is lowered when a greater steplength can be used with a lower order of the $BD$ formulas.

When the error term (8) contributes significantly to the principal local truncation error the condition of smoothness (10) is likely to be violated as $\theta$ is of the order of magnitude 1.

The preceding discussion leads to a test for instability which is sufficiently reliable for most purposes. The instability must be detected before the steplength control reduces the steplength, and the instability reaction should not be started unless the discretization actually is unstable.

**Instability test.**

During integration with a method of order $p$ the condition of smoothness (10) is taken for fulfilled when

$$||h^p y_n^{(p)}||_n > ||h^{p+1} y_n^{(p+1)}||_n.$$  

(11)

where $\sigma_p$ is found in table 1 for $p = 3, 4, 5, 6$.

A steady growth of the estimate of the local truncation error for fixed steplength and order can be expressed by the following growth relations:

$$||C_{p+1}h^{p+1}y_n^{(p+1)}|| > ||C_{p+1}h^{p+1}y_{n-1}^{(p+1)}|| > \ldots > ||C_{p+1}h^{p+1}y_{n-k}^{(p+1)}||.$$  

(12)
The discretization is considered unstable if the numerical solution \( y_n, y_{n-1}, \ldots \) violates the smoothness condition (11) and fulfils the growth relations (12).

Assume that a certain solution term dominates the local truncation error, and let the corresponding eigenvalue be \( \lambda_1 = \alpha_1 + i\omega_1 \). For a linear system of differential equations the principal local truncation error of the dominant term is

\[
\varepsilon_1 \approx C_{p+1}(h\lambda_1)^{p+1}b_1 \exp(\lambda_1x).
\]

The eigenvalue \( \lambda_1 \) may cause instability for \(|h\lambda_1| > 0.8\) approximately (cf. the regions of stability in [7]). For these magnitudes of \( h\lambda_1 \) the error in (13) is almost independent of the order \( p \) of the BD formula. This implies that the order can be lowered to improve the stability without significantly increasing the error contribution from \( b_1 \exp(h\lambda_1t) \). The decrease in order is limited by the remaining solution terms.

Experiments have shown that the following reaction to instability is appropriate.

When instability is detected during integration with a method of order \( p \) a new order \( q \) is chosen such that

\[
\|h^{q+1}y_n^{(q+1)}\| \leq \|h^r y_n^{(r)}\|, \quad r = 1, 2, \ldots, p.
\]

Two comments must be made on this instability reaction. First, the order is decreased at least by one, and second the new order \( q \) is possibly not the order giving the smallest truncation error as the error constant \( C_r \) decreases with increasing \( r \). The argument for this design of the instability reaction is the following. When instability is detected the estimate of the local truncation error is substantially greater than the corresponding truncation error of the smooth analytical solution. In the steps after change to a stable formula the instability errors are damped, and if \( \|C_{p+1}h^{q+1}y_n^{(q+1)}\| > \varepsilon \) it is reasonable to believe that the local truncation error after few steps is below the prescribed value \( \varepsilon \).

When the error measure of DIFSUB [6] is employed the precision requirement (9) may not be fulfilled, and the condition of smoothness (11) may be violated although the discretization is stable. In this case, however, the order is lowered because a greater steplength can be used with a lower order of the BD formula.

A growing and rapidly oscillating solution component will probably be caught by the instability test, but in this case the reduction in order will not cause the estimate of the local truncation error to diminish, on the contrary, and the steplength will soon be decreased. When the step-
length is smaller the numerical solution no longer appears rapidly oscillating, and the condition of smoothness (11) is fulfilled.

The error estimates provided by the Nordsieck vector are not applicable to the detection of instability. For a method of order $p$ the elements in the Nordsieck vector approximate the derivatives $(1/r!)|h^r y^{(r)}(x_n)$, $r = 1, 2, \ldots , p$ apart from the terms $c_{p+1}, h^{p+1} y^{(p+1)}(x_n)$, $r = 1, 2, \ldots , p$. This implies that the error estimate based on the numerical solution will be very bad when the numerical solution contains rapid oscillations due to instability.

A test for instability is described in [9]. The test is based on the detection of oscillations caused by instability. The changes of sign in the error estimate are monitored, and when the changes are too frequent precautions against instability are taken. The test is not very reliable for backward differentiation methods, and the described reactions to instability are only suited for non-stiff systems.

In the implementation described in [11] the order is chosen such that the terms in (3) have decreasing magnitudes. This is analogous to the requirement of smoothness (11). For constant steps the condition of smoothness in [11] can be formulated as $|h^p y^{(p)}| > |h^{p+1} y^{(p+1)}|$, and this condition is slightly milder than (11). This implies that the order control in [11] is able to handle problems with eigenvalues of the type $\lambda = \alpha \pm i\omega$, $|\omega/\alpha| \gg 1$.

A simplified version of the condition of smoothness (11) is used for instability detection in an implementation of the backward differentiation formulas described in [12]. A method of order $p$ is assumed to be unstable if $|h^p y^{(p)}| < |h^{p+1} y^{(p+1)}|$, and the order is switched to 1 as the instability reaction.

5. Algorithm for control of order and steplength.

The most important details of the algorithm for control of order and steplength are shown in Fig. 2. Some of the variables and parameters are used in the preceding sections, and the rest of the variables are explained by the algorithm. Details of the algorithm will be discussed below.

Order and steplength are constant for a number of steps, but the truncation error is monitored continuously, and a necessary reduction of the steplength is performed (lines 2, 3 and 4). The parameter $nc$ controls the intervals of constant order and steplength (lines 6 and 24).

The optimal order, i.e. the order permitting the greatest increase in steplength (or the smallest decrease), is found in lines 8-13. The inclusion of the swing factor $\gamma_s$ in line 11 may enforce a lower order than the
order with the smallest truncation error $\epsilon_{n,r}$. This improves the stability properties considerably.

The search for the optimal order is made among orders from 1 to $p+1$ (line 9). Experiments have shown, however, that only insignificant degradation of the algorithm occurs if the search is made among orders from $p-1$ to $p+1$.

The test for instability is made in line 14. If the logical expression is true the method is considered unstable. The formulas of order 1 and 2 are $A$-stable, and therefore $p > 2$ is necessary. The variable $\text{dec}$ counts the number of decreases in the error estimate during constant order and steplength (lines 5 and 24). The condition $\text{dec} \leq 1$ is somewhat arbitrary, but it has proved reasonable in practice. For the method of order 6 the value $s_n = 1$ is used, and violation of the condition of smoothness is taken as sufficient indication of instability.

1. $\text{next: compute solution point } y_n$;
2. $\epsilon_n := |C_{p+1}h_{n}^{p+1}y_n^{(p+1)}|$;
3. if $\epsilon_n > \epsilon_{\text{max}}$ then
4. begin $h_n := h_n(\epsilon/\epsilon_{n})^{1/(p+1)}$; go to next end;
5. if $\epsilon_n < \epsilon_{n-1}$ then $\text{dec} := \text{dec} + 1$;
6. if $nc \geq n$ then begin $h_{n+1} := h_n$; $x_{n+1} := x_n + h_{n+1}$;
7. $n := n + 1$; go to next end else
8. begin opt := 1;
9. for $r := 1$ step 1 until $p+1$ do
10. begin $\nabla_r := |C_{p+1}y_n^{(p+1)}|$; $\epsilon_{n,r} := C_{r+1} \nabla_r$;
11. $\gamma_r := \min[\epsilon_r, (\epsilon/\epsilon_{n,r})^{1/(p+1)}]$;
12. if $\gamma_r > \gamma_{\text{opt}}$ then $\text{opt} := r$;
13. end computation of optimal order;
14. if $p > 2 \land (\nabla_p/s_p) > \nabla_{p-1} \land (\text{dec} \leq 1 \land p = 6)$ then
15. begin stab := 1;
16. for $r := 1$ step 1 until $\min[\text{opt}, p-1]$ do
17. if $\nabla_r \leq \nabla_{\text{stab}}$ then stab := $r$;
18. opt := stab;
19. if $g_{\text{opt}} < 1$ then $g_{\text{opt}} := 1$;
20. end instability reaction;
21. $p := \text{opt}$;
22. if $g_p < 1.1 \land g_p > 0.9 \land g_p < \gamma_p$ then $g_p := 1$;
23. $h_{n+1} := g_p h_n$; $x_{n+1} := x_n + h_{n+1}$; $n := n + 1$;
24. $nc := n + p$; $\text{dec} := 0$; $\epsilon_{n-1} := 0$; go to next
25. end order and steplength control sequence;

Fig. 2.
When instability is detected a new order is found in lines 15-20. The order is lowered at least one (line 16), and the steplength is not allowed to decrease (line 19).

By permitting only significant changes in steplength a more reliable detection of instability is obtained. In line 22 a minimum change of 10% is required.

The only memory (from step to step) associated with the algorithm in Fig. 2 is the variable dec. For nonlinear systems of differential equations the conditions for the correct action of a memory may not be fulfilled. When the movement of the eigenvalues of the Jacobi matrix is modest advantage can be taken of the analogy with linear systems. In this case the following devise has proved useful.

When instability is detected for order \( p \) the maximum order is taken to be \( p - 1 \) for a number of steps. When the quarantine is taken off only 3 steps are permitted with order \( p \). If instability is detected again, 2 of the 3 steps are discarded. With this strategy the steps with the possibly instable formula will not introduce instability into the numerical solution.

As mentioned in the preceding discussions a number of the parameters are based on more or less heuristic arguments, and it is certainly possible to improve the performance of the integration program by changing some or all of these parameters. The set of parameters chosen here is only one among many possible. Also the structure of the algorithm for order and steplength control may be improved, as this structure is based on a simplified analysis of the solution of linear systems of differential equations.

6. Test examples.

The new algorithm for control of order and steplength (Fig. 2) was incorporated in an experimental implementation of the implicit backward differentiation methods. The corrector and predictor formulas are used just as in the definitions (1) and (2). During the change of steplength the coefficients are computed such that a method of order \( p \) is fitted to a polynomial of order \( p \). This approach is equivalent to the one discussed in section 2 C and in [3]. As mentioned in section 2 F a sort of absolute error measure and constant error per step is used in the error estimation and steplength control.

The efficiency of the instability detection and reaction was tested for a number of linear differential equations. A systematic test with equations of order 4 was carried out, and common to all systems were the eigenvalues \( \lambda_{1,2} = \pm i \). The remaining two eigenvalues \( \lambda_{3,4} = -\alpha \pm i\omega \)
were varied to study the efficiency of the integration method relative to the problem. The eigenvalues $\lambda_{1,2}$ correspond to sine and cosine terms, and the possibility of lowering the order for some steplength is therefore limited by the truncation error corresponding to the periodic solution terms. For the first class of problems $\omega/\pi = 20$, and instability is possible for the formulas of orders 3-6. For the second and the third set of problems, where $\omega/\pi = 10$ and 5 respectively, the formulas of orders 4-6 may be unstable.

The class of problems is defined in Fig. 3. The final value $x_f$ is chosen such that the exponentially decreasing terms are significant in approximately the first half of the integration interval. Instability appears when these terms are becoming insignificant. Therefore the main effort of the integration is concentrated in the first half of the integration interval.

$$
A = \text{diag}(i, -i, -\alpha + i\omega, -\alpha - i\omega)
$$

$$
y' = A A A^{-1} y, \quad x \in [0, x_f], \quad x_f = 20/\pi
$$

$$
A = \begin{pmatrix}
-1 & 1 & i & -i \\
1 & -i & -i & 1 \\
i & 1 & -1 & -1 \\
i & -i & 1 & 1
\end{pmatrix}, \quad y_0 = \begin{pmatrix}
-1 \\
1 \\
1 \\
-1
\end{pmatrix}
$$

$$
y(x) = \begin{pmatrix}
-\cos x - \exp(-\alpha x) \sin(\omega x) \\
\cos x - \exp(-\alpha x) \sin(\omega x) \\
-\sin x + \exp(-\alpha x) \cos(\omega x) \\
-\sin x - \exp(-\alpha x) \cos(\omega x)
\end{pmatrix}
$$

Fig. 3.

Table 3 shows the results of the test-runs. The method with instability check is denoted $MS$. For comparison the problems are solved again but without instability check. The only difference between method $MS$ and $ME$ is the bypassing of the instability test and reaction in $ME$.

The results in Table 3 show that the greatest improvement due to the instability test and reaction is obtained at the low precision for $\omega/\pi = 20$ and $\omega = 50$. As a general rule the greatest improvement is obtained for large values of $\omega/\pi$ and $\omega/\text{Im}[\lambda_1]$. In some cases the global error of the solution from method $ME$ is smaller than the global error from method $MS$. It is explained by the fact that the steplength of method $ME$ is limited by instability. Therefore the truncation error is much smaller than the required limit.
For $\omega/\alpha < 5$ the instability test is almost insignificant as the algorithm for computing the optimal order (lines 8-13 in Fig. 2) is able to decrease the order. As the exponential terms decrease the steplength can be increased, and when the increase is sufficiently rapid the limitation in swing (line 11) will force a lower order.

\[ \epsilon = 1E - 2 \]

<table>
<thead>
<tr>
<th>Problem</th>
<th>FCN Calls</th>
<th>No of steps</th>
<th>INV Calls</th>
<th>Global error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega/\alpha$</td>
<td>$\omega$</td>
<td>$MS$</td>
<td>$ME$</td>
<td>$MS$</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>420</td>
<td>1460</td>
<td>138</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>695</td>
<td>1471</td>
<td>256</td>
</tr>
<tr>
<td>5</td>
<td>307</td>
<td>1487</td>
<td>301</td>
<td>406</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>307</td>
<td>740</td>
<td>103</td>
</tr>
<tr>
<td>20</td>
<td>332</td>
<td>729</td>
<td>111</td>
<td>237</td>
</tr>
<tr>
<td>10</td>
<td>489</td>
<td>735</td>
<td>177</td>
<td>241</td>
</tr>
<tr>
<td>5</td>
<td>433</td>
<td>768</td>
<td>151</td>
<td>255</td>
</tr>
<tr>
<td>20</td>
<td>154</td>
<td>190</td>
<td>49</td>
<td>60</td>
</tr>
<tr>
<td>20</td>
<td>158</td>
<td>186</td>
<td>50</td>
<td>69</td>
</tr>
<tr>
<td>10</td>
<td>171</td>
<td>196</td>
<td>54</td>
<td>62</td>
</tr>
<tr>
<td>5</td>
<td>193</td>
<td>242</td>
<td>63</td>
<td>76</td>
</tr>
</tbody>
</table>

\[ \epsilon = 1E - 4 \]

<table>
<thead>
<tr>
<th>Problem</th>
<th>FCN Calls</th>
<th>No of steps</th>
<th>INV Calls</th>
<th>Global error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega/\alpha$</td>
<td>$\omega$</td>
<td>$MS$</td>
<td>$ME$</td>
<td>$MS$</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>1001</td>
<td>1721</td>
<td>345</td>
</tr>
<tr>
<td>20</td>
<td>1121</td>
<td>1714</td>
<td>400</td>
<td>576</td>
</tr>
<tr>
<td>10</td>
<td>1656</td>
<td>1703</td>
<td>583</td>
<td>572</td>
</tr>
<tr>
<td>5</td>
<td>1711</td>
<td>1718</td>
<td>602</td>
<td>576</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>611</td>
<td>949</td>
<td>202</td>
</tr>
<tr>
<td>20</td>
<td>673</td>
<td>948</td>
<td>236</td>
<td>310</td>
</tr>
<tr>
<td>10</td>
<td>809</td>
<td>928</td>
<td>278</td>
<td>304</td>
</tr>
<tr>
<td>5</td>
<td>832</td>
<td>864</td>
<td>310</td>
<td>311</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>439</td>
<td>480</td>
<td>143</td>
</tr>
<tr>
<td>20</td>
<td>418</td>
<td>479</td>
<td>139</td>
<td>155</td>
</tr>
<tr>
<td>10</td>
<td>420</td>
<td>482</td>
<td>143</td>
<td>156</td>
</tr>
<tr>
<td>5</td>
<td>473</td>
<td>502</td>
<td>161</td>
<td>168</td>
</tr>
</tbody>
</table>

Table 3.

The example of Fig. 3 is a very simple one, but it illustrates the sort of improvements that may be expected in practical problems where the eigenvalues of the Jacobi matrix are complex.
In [5] a number of available methods for solving stiff systems of differential equations are compared. One of these methods called GEAR is a modified version of the program DIFSUB in [6]. The program GEAR is efficient for all the problems in [5] except when some of the eigenvalues of the Jacobian are close to the imaginary axis ($\omega/x \gg 1$). Table 4 shows the results from three test problems for which GEAR was inefficient. The problems were also solved with MS (instability check) and with ME. The strategy for selecting steplength is somewhat different in GEAR and in MS and ME. It is therefore impossible to compare the methods MS and GEAR directly, but the results from the method ME indicate when instability control is essential. The test problems are listed in Fig. 4.

$$y' = \begin{pmatrix} -1 & 1 & 0 & 0 \\ -100 & -1 & 0 & 0 \\ 0 & 0 & -100 & 1 \\ 0 & 0 & -10000 & -100 \end{pmatrix} y, \quad y(0) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\lambda = (-1 \pm 10i, -100 \pm 1000i), \quad x \in [0, 20]$$

Problem B1

$$y' = \begin{pmatrix} -10 & \omega & 0 & 0 & 0 & 0 \\ -\omega & -10 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.1 \end{pmatrix} y, \quad y(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\lambda = (-0.1, -0.5, -1, -4, -10 \pm i\omega), \quad x \in [0, 20]$$

Problem B4: $\omega = 25$. Problem B5: $\omega = 100$.

Fig. 4.

The superior performance of method ME relative to GEAR is due to the algorithm for optimal order in ME. For moderate values of $\omega/x$ and moderate precision requirements this algorithm is able to reduce the order before instabilities arise.
Table 4.

7. Conclusion.

It was demonstrated how the instability test and reaction improved the performance of an implementation of backward differentiation formulas. The gain in efficiency is obtained when the eigenvalues of the Jacobi matrix are close to the imaginary axis.

Let $\lambda_r = \alpha_r + i\omega_r$ be an eigenvalue causing instability, and let $\lambda_{r+1} = \alpha_{r+1} + i\omega_{r+1}$ be the next eigenvalue where $|\alpha_{r+1}| < |\alpha_r|$. Then the improvement obtained by the instability test and reaction is large for a large ratio $|\omega_r|/|\omega_{r+1}|$ and small for a small ratio $|\omega_r|/|\omega_{r+1}|$.

Nonlinear problems were not included in the comparisons, but the instability test does not contain much memory, so the algorithm is expected to perform well for moderately nonlinear problems. For strongly nonlinear problems it may be advantageous to allow change of order and steplength after each step.

REFERENCES


**INST. FOR TELETEKNIK BYGNING 348**
**DANMARKS TEKNISKE HOJSKOLE**
**LYNGBY, DENMARK**