SIAM J. SCI. COMPUT.
Vol. 21, No. 6, pp. 2206–2224

ACCURACY OF DECOUPLED IMPLICIT INTEGRATION FORMULAS[*]

STIG SKELBOE[†]

Abstract. Dynamical systems can often be decomposed into loosely coupled subsystems. The system of ordinary differential equations (ODEs) modelling such a problem can then be partitioned corresponding to the subsystems, and the loose couplings can be exploited by special integration methods to solve the problem using a parallel computer or just solve the problem more efficiently than by standard methods.

This paper presents accuracy analysis of methods for the numerical integration of stiff partitioned systems of ODEs. The discretization formulas are based on the implicit Euler formula and the second order implicit backward differentiation formula (BDF2). Each subsystem of the partitioned problem is discretized independently, and the couplings to the other subsystems are based on solution values from previous time steps. Applied this way, the discretization formulas are called decoupled.

The stability properties of the decoupled implicit Euler formula are well understood. This paper presents error bounds and asymptotic error expansions to be used in controlling step size, relaxation between subsystems and the validity of the partitioning. The decoupled BDF2 formula is analyzed within the same framework.

Finally, the analysis is used in the design of a decoupled numerical integration algorithm with variable step size to control the local error and adaptive selection of partitionings. Two versions of the algorithm with decoupled implicit Euler and BDF2, respectively, are used in examples where a realistic problem is solved. The examples compare the results from the decoupled implicit Euler and BDF2 formulas, and compare them with results from the corresponding classical formulas.

Key words. Euler's implicit formula, backward differentiation formulas, multirate formulas, parallel numerical integration, partitioned systems, absolute stability

AMS subject classifications. 34-04, 34A65, 65L06, 65L70

PII. S1064827598337919

1. Introduction.

The numerical solution of stiff systems of ordinary differential equations (ODEs) requires implicit discretization formulas. The implicit algebraic problems are usually solved by a Newton-type iteration method which involves the solution of systems of linear equations that often turn out to be sparse. Attempts to parallelize a standard algorithm, as the one just outlined, often lead to disappointing efficiency because the parallel algorithm is too fine-grained or has too large a sequential fraction.

The waveform relaxation method [1], although not developed with parallel computation in mind, leads to efficient parallel algorithms for the class of problems where it works well. Multirate integration can be considered an integral feature of the waveform relaxation method. The relaxation part of the method is a potential source of computational inefficiency. One relaxation iteration is fairly expensive and convergence may be slow.

The decoupled integration methods in this paper were developed as a response to the problems encountered in parallelizing standard integration methods and the problems with waveform relaxation. The decoupled integration methods, like the waveform relaxation method, exploit a partitioning of a system of ODEs into loosely

[*] Received by the editors April 27, 1998; accepted for publication (in revised form) April 21, 1999; published electronically May 19, 2000.

http://www.siam.org/journals/sisc/21-6/33791.html

[†] Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark (stig@diku.dk).

coupled subsystems. The decoupled integration methods employ only one and occasionally two relaxation iterations. Multirate integration is not quite as natural as for the waveform relaxation method, although it is still possible, and the parallel implementations of decoupled integrations methods will be finer grained than parallel waveform relaxation methods. The decoupled integration methods may be more efficient on a sequential computer than standard integration methods, just like the waveform relaxation method.

A previous paper introduced the decoupled implicit Euler method [2]. The existence of a global error expansion was proved under very general choice of step size, thus permitting the use of Richardson extrapolation. A sufficient condition for stability of the discretization was given in [3]. This condition is called monotonic max-norm stability, and it guarantees contractivity. Partitioned systems of ODEs are in qualitative terms characterized as monotonically max-norm stable if each subsystem is stable and if the couplings from one subsystem to the others are weak.

This paper is organized as follows. Section 2, "Partitioned systems of ODEs and decoupled discretization formulas," gives preliminaries and definitions, including the definition of monotonic max-norm stability and the presentation of the decoupled implicit Euler and the decoupled implicit second order backward differentiation formula (BDF2).

Section 3, "Error bounds," gives error bounds for the classical and the decoupled implicit Euler formulas. The bounds are closely tied to the monotonic max-norm stability condition which applies only to the Euler formulas, and the bounds are not readily generalized to the BDF2 formula.

Section 4, "Asymptotic error formulas and error estimation," includes four subsections treating explicit formulas, classical implicit formulas, decoupled implicit Euler, and finally decoupled BDF2 formulas. The explicit formulas are used in error estimation and for prediction in the decoupled formulas. The asymptotic errors of the classical formulas are the smallest achievable for the decoupled formulas and therefore of interest. The asymptotic errors of the decoupled formulas are given for various modes of operation, and error estimation techniques are presented.

Section 5, "Integration algorithm," first presents general principles for decoupled integration algorithms derived from the previous analysis. Then the details of an implementation of the decoupled implicit Euler formula is given, followed by the minor modifications required for replacing the Euler formula with the BDF2 formula. The implementation employs variable step size to control the local error and adaptive selection of partitionings among two predetermined alternatives.

These integration algorithms are used in section 6, "Examples: Chemical reaction kinetics," where a real problem is solved using both decoupled and classical versions of implicit Euler and BDF2. The examples show excellent performance of the decoupled formulas.

2. Partitioned systems of ODEs and decoupled discretization formulas.

Define a system of ODEs,

$$Y' = F(t, Y),\ Y(t_0) = Y_0,\ \text{and } t \geq t_0, \tag{2.1}$$

where $Y:\ R \to R^S$, $F:\ R \times R^S \to R^S$, and F is Lipschitz continuous in Y. Stable systems of differential equations are considered stiff when the step size of the discretization by an *explicit* integration method is limited by stability of the discretization and not by accuracy. Efficient numerical integration of stiff systems therefore requires *implicit* integration methods.

Let the original problem (2.1) be partitioned as follows:

$$\begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_q' \end{pmatrix} = \begin{pmatrix} f_1(t,Y) \\ f_2(t,Y) \\ \vdots \\ f_q(t,Y) \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{pmatrix}, \quad Y(t_0) = \begin{pmatrix} y_{1,0} \\ y_{2,0} \\ \vdots \\ y_{q,0} \end{pmatrix}, \tag{2.2}$$

where $y_r : \ R \to R^{s_r}$, $f_r : \ R \times R^S \to R^{s_r}$, and $\sum_{i=1}^q \ s_i = S$. When necessary, the partitioning of Y will be stated explicitly as in $f_r(t, y_1, y_2, \ldots, y_q)$.

2.1. Monotonic max-norm stability.

The following stability condition introduced in [3] plays a crucial role for the stability of decoupled implicit integration methods.

Definition: Monotonic max-norm stability.

The partitioned system (2.2) is said to be monotonically max-norm stable if there exist norms $\| \cdot \|_r$ and functions $a_{rj}(t, U, V)$ such that

$$\begin{aligned} &\|u_r - v_r + \lambda \left[f_r(t,U) - f_r(t,V) \right] \|_r \\ &\qquad\qquad \geq \|u_r - v_r\|_r + \lambda \sum_{j=1}^q a_{rj}(t,U,V) \|u_j - v_j\|_j \end{aligned} \tag{2.3}$$

for all $t \geq t_0$, $\lambda \leq 0$, $U, V \in \Omega_t$ where $\Omega_t \subseteq R^S$, and the following condition holds for the logarithmic max-norm $\mu_\infty(\cdot)$ of the q × q matrix (a_{rj}):

$$\mu_\infty \left[(a_{rj}(t,U,V)) \right] \leq 0. \qquad \square \tag{2.4}$$

The condition (2.4) states that the matrix (a_{rj}) should be diagonally dominant with nonpositive diagonal elements. For a linear problem where $A_{rj} = \partial f_r / \partial y_j$, the (a_{rj}) matrix can be chosen as $a_{rj} = \|A_{rj}\|_p$ for $r \neq j$ and $a_{rr} = \mu_p(A_{rr})$.

Theorem 3 in [3] includes further results about monotonic max-norm stability, including possible choices of (a_{rj}). Monotonic max-norm stability admits arbitrarily stiff problems.

2.2. Decoupled implicit Euler: Stability, convergence.

The decoupled implicit Euler method is defined by the following discretization of the subsystems $r = 1, 2, \ldots, q$ by the implicit Euler formula [2]:

$$y_{r,n} = y_{r,n-1} + h_{r,n} f_r(t_{r,n}, \tilde{y}_{1,n}, \ldots, \tilde{y}_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \ldots, \tilde{y}_{q,n}), \tag{2.5}$$

where $n = 1, 2, \ldots,$ $t_{r,n} = t_0 + \sum_{j=1}^n h_{r,j}$, and the variables $\tilde{y}_{i,n}$ are convex combinations of values in $\{y_{i,k} \mid k \geq 0\}$ for $i \neq r$. The convex combinations $\tilde{y}_{i,n}$ will, in general, depend on subsystem index r, but in order to simplify notation this dependency will not be specified explicitly.

The method is called "decoupled" because the algebraic system resulting from the discretization of (2.1) by Euler's implicit formula is decoupled into a number of independent algebraic problems. The decoupled implicit Euler formula can be used as the basis of parallel methods where (2.5) is solved independently and in parallel for different r-values. The method can be used in multirate mode with $h_{r,n} \neq h_{j,n}$ for $r \neq j$, and the multirate formulation can be used in a parallel waveform relaxation method [1].

The sequential solution of (2.5) for $r = 1, 2, \ldots, q$ on a single processor will, in general, be computationally cheaper than solving the complete system $Y_n = Y_{n-1} + hF(t_n, Y_n)$. Therefore the decoupled implicit Euler method may also be an attractive alternative to the classical Euler formula on a sequential processor even when there is no multirate opportunity.

Theorems 4 and 5 in [3] assure the stability of the discretization by (2.5) of a monotonically max-norm stable problem and the convergence of waveform relaxation. The stability condition poses no restrictions on the choice of the step sizes $h_{r,n}$.

The convexity of $\tilde{y}_{r,n}$ is necessary in the stability theorem of the decoupled implicit Euler method. A convex combination would typically be a zero-order interpolation, $\tilde{y}_{r,n} = y_{r,n-1}$, except in multirate organizations, where a linear interpolation is an option.

The multirate aspect is discussed extensively in [2] and [3]. Most of the results and techniques in this paper can readily be extended to multirate algorithms. However, presenting results for multirate algorithms would add substantially to notational complexity, and furthermore, the algorithms and the example presented in sections 5 and 6, respectively, do not exploit multirate formulation. Therefore, the multirate aspect will not be explored further.

2.3. Organization: Jacobi, Gauss–Seidel, relaxation.

The definition of the decoupled implicit Euler formula in (2.5) suggests a Jacobi-type organization of the computation which is well suited for parallel computation.

Define the function $G_J(t, Y, \tilde{Y})$ from the partitioning of F given in (2.2),

$$g_{J,r}(t, Y, \tilde{Y}) = f_r(t, \tilde{y}_1, \ldots, \tilde{y}_{r-1}, y_r, \tilde{y}_{r+1}, \ldots, \tilde{y}_q), \tag{2.6}$$

where $g_{J,r} : R \times R^S \times R^S \to R^{s_r}$ and $\tilde{Y} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_q)$. Assuming the same step size h_n for all subsystems, the decoupled implicit Euler formula can now be expressed in the compact form,

$$Y_n = Y_{n-1} + h_n G_J(t_n, Y_n, \tilde{Y}_n). \tag{2.7}$$

The definition of G_J given above corresponds to the Jacobi organization of the computation. A similar G_{G-S} can be defined for the Gauss–Seidel organization,

$$g_{G-S,r}(t, Y, \tilde{Y}) = f_r(t, y_1, \ldots, y_{r-1}, y_r, \tilde{y}_{r+1}, \ldots, \tilde{y}_q). \tag{2.8}$$

The decoupled implicit Euler formula based on the Gauss–Seidel organization $Y_n = Y_{n-1} + h_n G_{G-S}(t_n, Y_n, \tilde{Y}_n)$ will, in general, be more accurate than (2.7). The Gauss–Seidel organization, although inherently sequential, may still be used in parallel if the partitioning (2.2) admits a red-black reordering or a similar reordering based on more colors (see, e.g., [4]). In the following the generic function G, meaning either G_J or G_{G-S}, will be used.

The decoupled implicit Euler formula can be used with relaxation iterations,

$$Y_n^{[m+1]} = Y_{n-1} + h_n G(t_n, Y_n^{[m+1]}, Y_n^{[m]}), \tag{2.9}$$

where $Y_n^{[0]} = \tilde{Y}_n$. The computational cost of each iteration of (2.9) is approximately the same as the cost of computing Y_n from (2.7).

If the relaxation is carried to convergence, the resulting discretization is the classical implicit Euler discretization.

When the numerical solution at t_n has been computed and accepted, it is usually used for computing the next step, and it is denoted by Y_n, where $Y_n := Y_n^{[m]}$.

In the relaxation (2.9), each subsystem r is solved in each sweep, usually by a Newton-type iteration carried to convergence. A relaxation iteration based on Newton's method with Jacobi or Gauss–Seidel organization as shown below may converge just as fast and with substantially less computation per iteration.

$$\begin{aligned} y_{r,n}^{[m+1]} = y_{r,n}^{[m]} - \left(I - \frac{\partial f_r}{\partial y_{r,n}}\right)^{-1} \Big(y_{r,n}^{[m]} - y_{r,n-1} \\ -h_n f_r \left(t_n, y_{1,n}^{[m+1]}, \dots, y_{r-1,n}^{[m+1]}, y_{r,n}^{[m]}, y_{r+1,n}^{[m]}, \dots, y_{q,n}^{[m]}\right)\Big) \end{aligned} \tag{2.10}$$

for $r = 1, 2, \dots, q$ and $y_{r,n}^{[0]} = \tilde{y}_{r,n}$. Again the potential for parallelization is less with Gauss–Seidel organization than with Jacobi organization.

2.4. Decoupled BDF2. The BDF2 can also be used in a decoupled mode,

$$Y_n = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n G(t_n, Y_n, \tilde{Y}_n), \tag{2.11}$$

where $\gamma_n = h_n/h_{n-1}$, $\alpha_1 = 1 - \alpha_2$, $\alpha_2 = -\gamma_n^2/(2\gamma_n + 1)$, and $\beta = (\gamma_n + 1)/(2\gamma_n + 1)$.

Stability results for this formula are only known for linear problems and constant step size [5].

The decoupled BDF2 can, of course, be relaxed just as the Euler formula in (2.9) or (2.10):

$$Y_n^{[m+1]} = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n G(t_n, Y_n^{[m+1]}, Y_n^{[m]}). \tag{2.12}$$

3. Error bounds.

3.1. Decoupled implicit Euler. The error of the decoupled implicit Euler formula can be bounded as follows. Consider the local truncation error for subsystem r,

$$\begin{aligned} \mathcal{L}[Y(t_n); h_n]_r &= y_r(t_n) - y_r(t_{n-1}) - h_n f_r(t_n, y_1(t_n), \dots, y_r(t_n), \dots, y_q(t_n)) \\ &= -\frac{h_n^2}{2} y_r^{(2)}(t_n) + \frac{h_n^3}{3!} y_r^{(3)}(t_n) - \cdots, \end{aligned}$$

and the decoupled implicit Euler formula using Jacobi organization,

$$y_{r,n} - y_r(t_{n-1}) - h_n f_r(t_n, \tilde{Y}_n^r) = 0,$$

where $\tilde{Y}_n^r = (\tilde{y}_{1,n}, \dots, \tilde{y}_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \dots, \tilde{y}_{q,n})$.

Assume that the monotonic max-norm stability condition (2.3), (2.4) is fulfilled with $Y(t_n), \tilde{Y}_n^r \in \Omega_{t_n}$, and introduce the simplified notation $a_{rj}^n = a_{rj}(t_n, Y(t_n), \tilde{Y}_n^r)$. Then subtraction leads to

$$\begin{aligned} \|\mathcal{L}[Y(t_n); h_n]_r\|_r &= \|y_r(t_n) - y_{r,n} - h_n[f_r(t_n, Y(t_n)) - f_r(t_n, \tilde{Y}_n^r)]\|_r \\ &\geq [1 - h_n a_{rr}^n]\|y_r(t_n) - y_{r,n}\|_r - h_n \sum_{j \neq r} a_{rj}^n \|y_j(t_n) - \tilde{y}_{j,n}\|_j. \end{aligned}$$

A bound for the local error $y_r(t_n) - y_{r,n}$ $(r = 1, 2, \dots, q)$ of the decoupled implicit Euler formula is then (cf. Lemma 2.2, Section III.2. in [6])

$$
\begin{aligned}
&\|y_r(t_n) - y_{r,n}\|_r \\
&\le [1 - h_n a^n_{rr}]^{-1}\left(\|\mathcal{L}[Y(t_n);h_n]_r\|_r + h_n \sum_{j\neq r} a^n_{rj}\|y_j(t_n) - \tilde{y}_{j,n}\|_j\right) \qquad (3.1)\\
&\le [1 - h_n a^n_{rr}]^{-1}\left(\|\mathcal{L}[Y(t_n);h_n]_r\|_r - h_n a^n_{rr} \max_{j\neq r}\|y_j(t_n) - \tilde{y}_{j,n}\|_j\right)\\
&= \theta_{r,n}\|\mathcal{L}[Y(t_n);h_n]_r\|_r + (1-\theta_{r,n})\max_{j\neq r}\|y_j(t_n) - \tilde{y}_{j,n}\|_j \qquad (3.2)
\end{aligned}
$$

since $\sum_{j\neq r}|a^n_{r,j}| \le -a^n_{rr}$ by (2.4) and $\theta_{r,n} = [1 - h_n a_{rr}(t_n, Y(t_n), \tilde{Y}^r_n)]^{-1}$, where $0 < \theta_{r,n} \le 1$.

3.2. Classical implicit Euler. A similar local error bound can be established for the classical implicit Euler formula expressed as follows in a notation analogous to (2.5):

$$y_{r,n} = y_r(t_{n-1}) + h_n f_r(t_n, y_{1,n}, \ldots, y_{r-1,n}, y_{r,n}, y_{r+1,n}, \ldots, y_{q,n}).$$

Using the monotone max-norm stability condition, we obtain

$$
\begin{aligned}
L(t_n) &\ge \left\| [I - h_n(a_{ij}(t_n, Y(t_n), Y_n))] \begin{pmatrix} \|y_1(t_n) - y_{1,n}\|_1 \\ \|y_2(t_n) - y_{2,n}\|_2 \\ \vdots \\ \|y_q(t_n) - y_{q,n}\|_q \end{pmatrix} \right\|_\infty \\
&\ge (1 - h_n \mu_\infty[(a_{ij}(t_n, Y(t_n), Y_n))])\max_r \|y_r(t_n) - y_{r,n}\|_r \\
&\ge (1 - h_n m_\infty(t_n))\max_r \|y_r(t_n) - y_{r,n}\|_r,
\end{aligned}
$$

where $L(t_n) = \max_r \|\mathcal{L}[Y(t_n);h_n]_r\|_r$, $Y(t_n), Y_n \in \Omega_{t_n}$, and $m_\infty(t)$ is defined as

$$m_\infty(t) = \sup_{U,V\in\Omega_t} \mu_\infty[(a_{ij}(t,U,V))].$$

A bound for the local error $y_r(t_n) - y_{r,n}$ $(r = 1, 2, \ldots, q)$ of the classical implicit Euler formula analogous to (3.2) is then

$$\|y_r(t_n) - y_{r,n}\|_r \le [1 - h_n m_\infty(t_n)]^{-1} L(t_n), \qquad (3.3)$$

where $0 < \theta_{r,n} \le [1 - h_n m_\infty(t_n)]^{-1} \le 1$. Both (3.2) and (3.3) are valid for all values of step sizes but may be most interesting and useful for values where $L(t_n) \approx \|\frac{h_n^2}{2} Y''(t_n)\|$.

The main difference between (3.2) and (3.3) is the last term in (3.2). The local truncation error norm $L(t_n)$ is $\mathcal{O}(h_n^2)$. The order of the last term in (3.2) is $\mathcal{O}(h_n^2)$ since $\max_{j\neq r}\|y_j(t_n) - \tilde{y}_{j,n}\|_j = \mathcal{O}(h_n)$ for $\tilde{y}_{j,n} = y_j(t_{n-1})$ and $1 - \theta_{r,n} = \mathcal{O}(h_n)$ for $h_n \to 0$. If the off-diagonal elements of (a_{ij}) are very small, the last term of (3.1) may be negligible, and the local error bound of the decoupled implicit Euler formula is almost equal to the local error bound of the classical implicit Euler formula.

4. Asymptotic error formulas and error estimation.

4.1. Explicit formulas. The numerical solution of stiff systems of ODEs requires implicit discretization formulas. A numerical integration algorithm typically also includes an explicit formula to be used in error estimation and for computing an initial value for the iterative (Newton-type) method used in solving the implicit discretization problem. The decoupled implicit Euler and BDF2 formulas furthermore require the computation of $\tilde{Y}_n$, where the use of an explicit formula is an option.

The explicit Euler formula $Y_n^e = Y_{n-1} + hF(t_{n-1}, Y_{n-1})$ is an obvious choice in connection with the implicit Euler formula, $Y_n = Y_{n-1} + hF(t_n, Y_n)$, as well as with the implicit decoupled Euler formula. However, explicit formulas including $F(t, Y)$ should generally be avoided in connection with stiff problems when $h\|\partial F/\partial Y\| \gg 1$. The bound $\|Y_n^e - Y_n\| \le h\|\partial F/\partial Y\|\|Y_{n-1} - Y_n\|$ may be approached if Y_{n-1} is off the smooth solution, which is the case if $Y_{n-1} = Y_{n-2} + hF(t_{n-1}, Y_{n-1})$ is only solved approximately for Y_{n-1}. Therefore polynomial interpolation formulas are preferred as predictors [7].

The linear interpolation formula,

$$Y_n^p = Y_{n-1} + \gamma_n(Y_{n-1} - Y_{n-2}), \tag{4.1}$$

where $\gamma_n = h_n/h_{n-1}$, has the local error expansion

$$Y(t_n) - Y_n^p = \frac{h_n^2}{2}(1 + \gamma_n^{-1})Y''(t_n) + \mathcal{O}(h_n^3)$$

for $Y_{n-1} = Y(t_{n-1})$ and $Y_{n-2} = Y(t_{n-2})$.

The second order polynomial predictor formula is

$$Y_n^{p2} = \bar{\alpha}_1 Y_{n-1} + \bar{\alpha}_2 Y_{n-2} + \bar{\alpha}_3 Y_{n-3}, \tag{4.2}$$

$$\bar{\alpha}_1 = 1 - \bar{\alpha}_2 - \bar{\alpha}_3, \quad \bar{\alpha}_2 = \frac{\gamma_n(\gamma_n + \delta_n)}{1 - \delta_n}, \quad \bar{\alpha}_3 = \frac{\gamma_n(\gamma_n + 1)}{\delta_n(\delta_n - 1)},$$

where $\gamma_n = h_n/h_{n-1}$ and $\delta_n = 1 + h_{n-2}/h_{n-1}$.

The local error expansion for this formula is

$$Y(t_n) - Y_n^{p2} = \bar{C}_3 h_n^3 Y^{(3)}(t_n) + \mathcal{O}(h_n^4), \quad \text{where } \bar{C}_3 = \frac{1}{6}(1 + \gamma_n^{-3}(\bar{\alpha}_2 + \bar{\alpha}_3\delta_n^3)),$$

assuming that $Y_{n-1} = Y(t_{n-1})$, $Y_{n-2} = Y(t_{n-2})$, and $Y_{n-3} = Y(t_{n-3})$.

4.2. Classical implicit Euler and BDF2. With the classical implicit Euler formula

$$Y_n = Y_{n-1} + h_n F(t_n, Y_n), \tag{4.3}$$

the local error $Y(t_n) - Y_n$ where $Y_{n-1} = Y(t_{n-1})$ can be expressed by

$$Y(t_n) - Y_n = h_n e_1 + h_n^2 e_2 + h_n^3 e_3 + \cdots \tag{4.4}$$

for

$$e_1 = 0, \quad e_2 = -\frac{1}{2}Y''(t_n), \quad e_3 = \frac{1}{6}Y^{(3)}(t_n) - \frac{1}{2}\frac{\partial F}{\partial Y}Y''(t_n),$$

where all partial derivatives above and in the rest of section 4 are computed at $Y(t_n)$.

The e-terms are obtained by substituting the expansion for Y_n (4.4) into the Euler formula (4.3), Taylor expanding $Y(t_{n-1})$ and $F(t_n, Y_n)$ at the point $(t_n, Y(t_n))$, and finally identifying e-terms of equal power of h_n.

The principal local error term $h_n^2 e_2 = -\frac{1}{2} h_n^2 Y''(t_n)$ can be estimated as

$$-\frac{1}{2} h_n^2 Y''(t_n) \approx (Y_n^p - Y_n)/(1 + \gamma_n^{-1}) = -h_n^2 Y_n[t_{n-2}, t_{n-1}, t_n], \tag{4.5}$$

where $Y_n[t_{n-2}, t_{n-1}, t_n]$ is the divided difference of the values Y_{n-2}, Y_{n-1}, Y_n.

The classical BDF2 formula (cf. (2.11)),

$$Y_n = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n F(t_n, Y_n),$$

has the local error expansion

$$Y(t_n) - Y_n = C_3 h_n^3 Y^{(3)}(t_n) + \mathcal{O}(h_n^4), \quad \text{where } C_3 = \frac{1}{6}(1 - 3\beta + \alpha_2 \gamma_n^{-3}), \tag{4.6}$$

assuming that $Y_{n-1} = Y(t_{n-1})$ and $Y_{n-2} = Y(t_{n-2})$.

The principal local error term of BDF2 can be estimated from

$$C_3 h_n^3 Y^{(3)}(t_n) \approx (C_3/\bar{C}_3)(Y_n - Y_n^{p2}) = 6 C_3 h_n^3 Y_n[t_{n-3}, t_{n-2}, t_{n-1}, t_n]. \tag{4.7}$$

The error estimates (4.5) and (4.7), based on divided differences, are asymptotically correct [8] if the step size varies according to $h_n = h\phi(t_{n-1})$, $0 < \Delta \leq \phi(t) \leq 1$, and $\phi(t)$ is sufficiently smooth. This is essentially the step size variation attempted by the algorithms in section 5.2.

4.3. Decoupled implicit Euler.

The local error of the decoupled implicit Euler formula (2.7) depends on the definition of $\tilde{Y}_n$ and the number of relaxation iterations being performed (cf. (2.9)). Two different modes corresponding to different choices of $\tilde{Y}_n$ are considered for different number of relaxation iterations.

If the partitioned system (2.2) is monotonically max-norm stable, then mode 1 discretizations are stable while the mode 2 discretizations cannot be guaranteed to be stable.

4.3.1. Mode 1, $\tilde{Y}_n = Y_{n-1}$.

The local error is expressed as follows:

$$Y(t_n) - Y_n^{[m]} = h_n e_1^{[m]} + h_n^2 e_2^{[m]} + h_n^3 e_3^{[m]} + \cdots. \tag{4.8}$$

The $e_r^{[m]}$-terms of (4.8) are found analogously to the e_r-terms of (4.4) from (2.7), $Y_n^{[1]} = Y_{n-1} + h_n G(t_n, Y_n^{[1]}, Y_{n-1})$ (cf. (2.9)),

$$e_1^{[1]} = 0, \quad e_2^{[1]} = -\frac{1}{2} Y''(t_n) + \frac{\partial G}{\partial \tilde{Y}} Y'(t_n),$$
$$e_3^{[1]} = \frac{1}{6} Y^{(3)}(t_n) - \frac{1}{2} \frac{\partial F}{\partial Y} Y''(t_n) + \frac{\partial G}{\partial Y} \frac{\partial G}{\partial \tilde{Y}} Y'(t_n) - \frac{1}{2} \frac{\partial^2 G}{\partial \tilde{Y}^2} Y'(t_n)^2.$$

The relation $(\partial G/\partial Y) + (\partial G/\partial \tilde{Y}) = \partial F/\partial Y$ was exploited in the expression for $e_3^{[1]}$. Element i of the vector $(\partial^2 G/\partial \tilde{Y}^2) Y'(t_n)^2$ is evaluated as $Y'(t_n)^T (\partial^2 G_i/\partial \tilde{Y}^2) Y'(t_n)$.

The principal local error term $h_n^2 e_2^{[1]}$ can only be estimated using $Y_n^p - Y_n^{[1]}$ (cf. (4.5)) if $\|\frac{1}{2} Y''(t_n)\| \gg \|(\partial G/\partial \tilde{Y}) Y'(t_n)\|$. This inequality may be fulfilled when the subsystems are loosely coupled, but it is by no means guaranteed by the monotonic max-norm condition. Error estimation based on $Y_n^p - Y_n^{[1]}$ should only be used if the quality of this estimate is somehow monitored continually.

The $e_r^{[2]}$-terms of (4.8) are found from $Y_n^{[2]} = Y_{n-1} + h_n G(t_n, Y_n^{[2]}, Y_n^{[1]})$ like the analogous terms above:

$$e_1^{[2]} = 0, \quad e_2^{[2]} = -\frac{1}{2} Y''(t_n),$$
$$e_3^{[2]} = \frac{1}{6} Y^{(3)}(t_n) - \frac{1}{2} \frac{\partial F}{\partial Y} Y''(t_n) + \left(\frac{\partial G}{\partial \tilde{Y}} \right)^2 Y'(t_n).$$

The relation $(\partial G / \partial Y) + (\partial G / \partial \tilde{Y}) = \partial F / \partial Y$ was exploited in the expression for $e_3^{[2]}$.

For $m \geq 2$, $e_2^{[m]} = e_2$ of the classical implicit Euler formula. Therefore the principal local error term $h_n^2 e_2^{[m]}$ can be estimated using (4.5) with $Y_n^{[m]}$ substituted for Y_n.

The $e_1^{[m]}$ and $e_2^{[m]}$ terms are unchanged for further relaxation iterations, $m \geq 3$, and

$$e_3^{[m]} = \frac{1}{6} Y^{(3)}(t_n) - \frac{1}{2} \frac{\partial F}{\partial Y} Y''(t_n) \quad \text{for} \quad m \geq 3,$$

which is identical to e_3 for the classical implicit Euler formula.

4.3.2. Mode 2, $\tilde{Y}_n = Y_n^p$. The local error is expressed as follows:

$$Y(t_n) - \bar{Y}_n^{[m]} = h_n \bar{e}_1^{[m]} + h_n^2 \bar{e}_2^{[m]} + h_n^3 \bar{e}_3^{[m]} + \cdots.$$

The $\bar{e}_1^{[1]}$- and $\bar{e}_2^{[1]}$-terms for $\bar{Y}_n^{[1]} = Y_{n-1} + h_n G(t_n, \bar{Y}_n^{[1]}, Y_n^p)$ are identical to e_1 and e_2 (classical implicit Euler), respectively, while

$$\bar{e}_3^{[1]} = \frac{1}{6} Y^{(3)}(t_n) - \frac{1}{2} \frac{\partial F}{\partial Y} Y''(t_n) + \frac{1}{2}(1 + \gamma_n^{-1}) \frac{\partial G}{\partial \tilde{Y}} Y''(t_n).$$

The computational cost of Y_n^p is, in general, much less than the cost of $Y_n^{[1]}$ in mode 1, and although they lead to different e_3-expressions, $\bar{e}_3^{[1]} \neq e_3^{[2]}$, one value will not be smaller than the other in general. However, the use of $\tilde{Y}_n = Y_n^p$ instead of $\tilde{Y}_n = Y_{n-1}$ may compromise the stability of the decoupled implicit Euler formula; cf. section 2.2.

As in mode 1, $\bar{e}_1^{[m]}$ and $\bar{e}_2^{[m]}$ are unchanged by further relaxation iterations, $m \geq 2$ and $\bar{e}_3^{[m]} = e_3$ (classical implicit Euler) for $m \geq 2$.

The development of the error terms $e_3^{[m]}$ and $\bar{e}_3^{[m]}$ is given to illustrate the influence of increasingly accurate values of $Y_n^{[m]}$. If the decoupling of the original problem into subsystems is efficient, mode 1 or 2 of the decoupled implicit Euler formula gives results very close to those of the classical implicit Euler formula and

$$\|Y_n^{[m]} - Y_n\| \ll \|\tilde{Y}_n - Y_n^{[m]}\| \quad \text{for} \quad m = 1 \quad \text{or} \quad m = 2. \tag{4.9}$$

If the decoupling is poor, the differences in the higher order $e_r^{[m]}$- or $\bar{e}_r^{[m]}$-terms will be significant and $\|Y_n^{[m]} - Y_n\| \approx \|\tilde{Y}_n - Y_n^{[m]}\|$.

The error expansions do not include multirate integration. The derivation of error expansions analogous to the above covering multirate integration could have been done using the basic definition (2.5), but it would be notationally complicated, and the results are essentially the same.

4.4. Decoupled BDF2. The decoupled BDF2 (2.11) admits more "natural" choices for $\tilde{Y}_n$ than the decoupled implicit Euler formula. Three modes using increasingly accurate $\tilde{Y}_n$ are presented.

Mode 1 is expected to possess the best stability properties among the three different modes although few theoretical results are available [5]. Mode 3 with its second order predictor value for $\tilde{Y}_n$ is expected to be the weakest in terms of stability properties, while mode 2 is somewhere in between.

4.4.1. Mode 1, $\tilde{Y}_n = Y_{n-1}$. The errors of the decoupled BDF2 are considered for

$$Y_n^{[1]} = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n G(t_n, Y_n^{[1]}, Y_{n-1}) \tag{4.10}$$

and subsequent relaxation iterations (2.12). The local error is expressed as follows for constant step size,

$$Y(t_n) - Y_n^{[m]} = h e_1^{[m]} + h^2 e_2^{[m]} + h^3 e_3^{[m]} + \cdots . \tag{4.11}$$

The $e_r^{[m]}$-terms of (4.11) are found analogously to the e_r-terms of (4.4):

$$e_1^{[1]} = 0, \quad e_2^{[1]} = \frac{2}{3}\frac{\partial G}{\partial \tilde{Y}} Y'(t_n),$$
$$e_3^{[1]} = -\frac{2}{9} Y^{(3)}(t_n) - \frac{1}{3}\frac{\partial G}{\partial \tilde{Y}} Y''(t_n) + \frac{4}{9}\frac{\partial G}{\partial Y}\frac{\partial G}{\partial \tilde{Y}} Y'(t_n) - \frac{1}{3}\frac{\partial^2 G}{\partial \tilde{Y}^2} Y'(t_n)^2.$$

The full order of accuracy of the BDF2 is not reached so another relaxation iteration is performed:

$$Y_n^{[2]} = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n G(t_n, Y_n^{[2]}, Y_n^{[1]}). \tag{4.12}$$

The error terms are now $e_1^{[2]} = 0$, $e_2^{[2]} = 0$, and

$$e_3^{[2]} = -\frac{2}{9} Y^{(3)}(t_n) + \frac{4}{9}\left(\frac{\partial G}{\partial \tilde{Y}}\right)^2 Y'(t_n).$$

If $\|Y^{(3)}(t_n)\| \gg \|2(\partial G/\partial \tilde{Y})^2 Y'(t_n)\|$, then the principal local error term can be estimated using formula (4.7). The inequality may be fulfilled when the subsystems are loosely coupled, but it is not guaranteed by the monotonic max-norm stability condition.

Since the error resulting from (4.10) is $\mathcal{O}(h^2)$, this step might be replaced by the decoupled implicit Euler formula mode 1, the result of which is denoted by $Y_n^{e1[1]}$ in this subsection. The BDF2 relaxation (4.12) is then replaced by

$$\hat{Y}_n^{[2]} = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n G(t_n, \hat{Y}_n^{[2]}, Y_n^{e1[1]}).$$

The local error expansion for constant step size is

$$Y(t_n) - \hat{Y}_n^{[2]} = -\frac{2}{9} h^3 \left(Y^{(3)}(t_n) + \frac{3}{2}\frac{\partial G}{\partial \tilde{Y}} Y''(t_n) - 3\left(\frac{\partial G}{\partial \tilde{Y}}\right)^2 Y'(t_n)\right) + \mathcal{O}(h^4).$$

The decoupled Euler–BDF2 combination is expected to be the $\mathcal{O}(h^3)$-error decoupled formula with the best stability properties. This expectation is based on the fact that the decoupled implicit Euler formula mode 1 is stable when the partitioned system (2.2) is monotonically max-norm stable. Such a result does not exist for the decoupled BDF2 formula (4.10).

Yet another relaxation iteration from $Y_n^{[2]}$ or $\hat{Y}_n^{[2]}$,

$$Y_n^{[3]} = \alpha_1 Y_{n-1} + \alpha_2 Y_{n-2} + \beta h_n G(t_n, Y_n^{[3]}, Y_n^{[2]}),$$

leads to a local error expansion with the same principal local error term as the classical BDF2 (4.6), including the case of variable step size. The principal local error term can thus be estimated using formula (4.7).

The computational cost of using mode 1 is rather high, so therefore, the following modes are of practical interest.

4.4.2. Mode 2, $\tilde{Y}_n = Y_n^p$. Another possible choice for $\tilde{Y}_n$ is $\tilde{Y}_n = Y_n^p$ computed from (4.1). The corresponding local error expansion for constant step size is

$$Y(t_n) - \bar{Y}_n^{[1]} = -\frac{2}{9} h^3 \left(Y^{(3)}(t_n) - 3 \frac{\partial G}{\partial \tilde{Y}} Y''(t_n) \right) + \mathcal{O}(h^4),$$

assuming that $Y_{n-1} = Y(t_{n-1})$ and $Y_{n-2} = Y(t_{n-2})$. The principal local error term can be estimated using formula (4.7) when the subsystems are loosely coupled.

Another relaxation iteration would lead to the same principal local error as for the classical BDF2 so that the principal local error term can be estimated using (4.7).

4.4.3. Mode 3, $\tilde{Y}_n = Y_n^{p2}$. Using this mode we obtain the same principal local error term as for the classical BDF2, and therefore, we also have the same possibility of estimating the error using formula (4.7).

The use of $\tilde{Y}_n = Y_n^{p2}$ with the decoupled BDF2 may give rise to some concern about the stability of the discretization.

4.4.4. Summary of decoupled BDF2 formulas. The following table summarises the local error results of the decoupled BDF2 formula. Any combination of mode and number of relaxations having the error $\mathcal{O}(h_n^3)$ will have the principal local error $C_3 h_n^3 Y_n^{(3)}$ after one or more additional relaxations.

Mode	1	1	1	1	2	2	3
$\tilde{Y}_n$	Y_{n-1}	Y_{n-1}	Y_{n-1}	$Y_n^{e1[1]}$	Y_n^p	Y_n^p	Y_n^{p2}
Error	$\mathcal{O}(h_n^2)$	$\mathcal{O}(h_n^3)$	$C_3 h_n^3 Y_n^{(3)}$	$\mathcal{O}(h_n^3)$	$\mathcal{O}(h_n^3)$	$C_3 h_n^3 Y_n^{(3)}$	$C_3 h_n^3 Y_n^{(3)}$
Relax'ns	1	2	3	2	1	2	1

5. Integration algorithm.

5.1. General principles.

The previous sections have presented the decoupled implicit Euler formula (section 2.2) and various iteration techniques for approaching the classical implicit Euler formula (section 2.3). Asymptotic local error expansions have been given for different modes of employment and corresponding local error estimation techniques (section 4.3). Finally, similar results have been presented for the decoupled BDF2 (sections 2.4 and 4.4).

All of these components can be used to construct numerous integration algorithms where the design decisions may be guided by the properties of the problem to be solved. The main objective of an integration algorithm is to solve a problem to a specified accuracy using as few arithmetic operations as possible.

The following discussion only deals with the decoupled implicit Euler formula since the stability results and error bound only apply to this formula. The BDF2 formula is expected to have analogous properties, and the implementation of the decoupled BDF2 formula is very similar to the implementation of the decoupled Euler formula.

5.1.1. Mode. The local error bound (3.1) shows how an accurate value of $\tilde{Y}_n$ reduces the influence of the partitioning on the local error. According to section 4.3, mode 2 is to be preferred over mode 1 because of the accuracy of $\tilde{Y}_n = Y_n^p$ which can be computed at little additional cost. Although $\|Y_n^p - Y(t_n)\|$ is smaller than $\|Y_{n-1} - Y(t_n)\|$ for $h_n \to 0$, the reverse may be true for larger values of h_n. Mode 1 should therefore be preferred if $\|Y_{n-1} - Y(t_n)\| < \|Y_n^p - Y(t_n)\|$.

An alternative approach for improving the accuracy of $\tilde{Y}_n$ is relaxation (2.9). After one relaxation iteration, $\tilde{Y}_n$ can be considered having the value $\tilde{Y}_n = Y_n^{[1]}$, etc. Relaxation does not increase the mode, and it is attractive in this respect. However, relaxation is computationally expensive and should only be used when it is strictly necessary.

5.1.2. Partitioning error. The error due to the partitioning is described by the matrix (a_{rj}). An aggressive partitioning with few small subsystems and otherwise scalar equations may lead to an (a_{rj}) matrix with relatively large off-diagonal elements, and it may not be diagonally dominant (2.4). According to (3.1), the error term including $\|Y(t_n) - \tilde{Y}_n\|$ may therefore contribute significantly.

A conservative partitioning will typically have some larger subsystems to accommodate strong couplings and to assure numerically small off-diagonal elements in (a_{rj}). The error bound (3.1) clearly shows how this may lead to a decoupled Euler formula with essentially the same error properties as the classical formula.

The Gauss–Seidel organization (2.8) takes advantage of a nonsymmetric structure of (a_{rj}). Assume that the partitioned system (2.2) is reordered symmetrically in equation number and variable number to make (a_{rj}) as close to lower triangular as possible, $\|(a_{rj})_{r>j}\| \gg \|(a_{rj})_{r<j}\|$. Then (3.1) is modified as follows,

$$\begin{aligned}\|y_r(t_n) - y_{r,n}\|_r \le (1 - h_n a_{rr}^n)^{-1}(\|\mathcal{L}[Y(t_n); h_n]_r\|_r \\ + h_n \sum_{j<r} a_{rj}^n \|y_j(t_n) - y_{j,n}\|_j + h_n \sum_{j>r} a_{rj}^n \|y_j(t_n) - \tilde{y}_{j,n}\|_j),\end{aligned}$$

and the larger lower triangular a_{rj}^n-values are multiplied with the smaller $\|y_j(t_n) - y_{j,n}\|_j$ errors, while the smaller upper triangular a_{rj}^n-values are multiplied with the larger $\|y_j(t_n) - \tilde{y}_{j,n}\|_j$ errors.

5.1.3. Stability. Stability is assured by the matrix (a_{rj}) being diagonally dominant (2.4) and by $\tilde{Y}_n$ being a convex combination of previous solution values. In mode 1, where $\tilde{Y}_n = Y_{n-1}$, the latter condition is fulfilled. The diagonal dominance condition may be fulfilled by a sufficiently conservative partitioning.

However, the monotonic max-norm stability condition in section 2.1 is a sufficient condition but not a *necessary* condition. Therefore, mode 2 may be used without encountering stability problems and also used when the diagonal dominance condition (2.4) is not fulfilled. A more conservative partitioning where (2.4) is fulfilled or closer to being fulfilled will not only improve stability but most likely also accuracy; cf. the previous section on partitioning error. However, a conservative partitioning leads to a more computationally expensive discretization than a more aggressive partitioning.

Relaxation iterations in mode 1 do not compromise stability, and furthermore, the monotonic max-norm stability guarantees convergence of the process.

5.2. Implementation details. The algorithm will use the decoupled implicit Euler formula or BDF2 with variable step size and choose between two different partitionings: an aggressive partitioning and a conservative partitioning.

The aggressive partitioning uses the smallest subsystems possible in order to minimise computational cost. The conservative partitioning uses somewhat larger subsystems in order to maintain accuracy during transient solution phases.

The decoupled implicit Euler formula is used in mode 2 ($\tilde{Y}_n = Y_n^p$) with one relaxation iteration ($Y_n := \bar{Y}_n^{[1]}$) except when $\|Y_{n-1} - Y_{n-1}^p\| > \|Y_{n-1} - Y_{n-2}\|$. Then mode 1 ($\tilde{Y}_n = Y_{n-1}$) is used with two relaxation iterations ($Y_n := Y_n^{[2]}$).

The quality of the partitioning is monitored using (4.9). The classical Euler solution should not be computed because of the incurred cost. In mode 1, Y_n in the partitioning criterion above is therefore replaced by $Y_n^{[2]}$,

$$\|Y_n^{[1]} - Y_n^{[2]}\| / \|\tilde{Y}_n - Y_n^{[1]}\| < \sigma,$$

for some $\sigma < 1$. In mode 2, $Y_n^{[1]}$ and $Y_n^{[2]}$ are replaced by $\bar{Y}_n^{[1]}$ and $\bar{Y}_n^{[2]}$, respectively.

Since mode 1 is used with two relaxation iterations, $Y_n^{[2]}$ is always available, and the cost involved in monitoring the partitioning is negligible.

Mode 2 is used with just one relaxation iteration. Therefore, the cost of a step where the partitioning is monitored is double the cost of an ordinary mode 2 step since $\bar{Y}_n^{[2]}$ is required. When $\bar{Y}_n^{[2]}$ is available in mode 2, it seems obvious to return $\bar{Y}_n^{[2]}$ as the result of a step, $Y_n := \bar{Y}_n^{[2]}$, since $\bar{Y}_n^{[2]}$ supposedly is more accurate than $\bar{Y}_n^{[1]}$. However, $Y_n := \bar{Y}_n^{[2]}$ is only returned occasionally, with $Y_n := \bar{Y}_n^{[1]}$ being the common result, and in the following example this generates oscillations in the local error estimate while failing to improve accuracy. Therefore, $Y_n := \bar{Y}_n^{[1]}$ is always returned from mode 2.

The integration algorithm can now be outlined as follows.

Initialization, step 1:

- choose *conservative partitioning*, $N_{monitor} = 2$
- $h_1 = h_{init}$
- no error estimation ($h_2 = h_1$)

Step n:

- compute Y_n^p from (4.1) and compute $\bar{Y}_n^{[1]}$ from the decoupled backward Euler formula using mode 2 ($Y_n^{[2]}$ computed by mode 1 is only used when the predicted solution Y_n^p is inaccurate, as explained above)
- **if** $n = N_{monitor}$ **then** *Monitor partitioning*
- estimate the principal local error term using (4.5), $\varepsilon_{est} = \|Y_n^p - Y_n\| / (1 + \gamma_n^{-1})$
- new step size, $h_{n+1} = \frac{h_n}{2}(1 + \sqrt{\varepsilon_{tol}/\varepsilon_{est}})$
- **if** $(h_{n+1} < h_n) \quad \wedge \quad (N_{monitor} > n + 1) \quad \wedge$ (*aggressive partitioning*) **then** $N_{monitor} = N_{monitor} - 1$

The step size formula averages 1 and $\sqrt{\varepsilon_{tol}/\varepsilon_{est}}$ to reduce the tendency of oscillations in step size selection.

If the step size is decreasing, the solution may be entering a transient phase which requires the conservative partitioning. If the aggressive partitioning is being used, the number of steps until the next *Monitor partitioning* is reduced by decreasing $N_{monitor}$.

The partitioning is chosen using the following conditions.

Monitor partitioning

- in mode 2, relax the decoupled implicit Euler formula an extra time to compute $\bar{Y}_n^{[2]}$. In mode 1, use $Y_n^{[1]}$ and $Y_n^{[2]}$ in the following test.
- **if** $\|\bar{Y}_n^{[1]} - \bar{Y}_n^{[2]}\| / \|Y_n^p - \bar{Y}_n^{[1]}\| < 0.6$ **then** choose *aggressive partitioning* **else** choose *conservative partitioning*
- **if** shift from *conservative* to *aggressive partitioning* **then** $N_{monitor} = n + 1$ **else** $N_{monitor} = n + 10$

A switch from aggressive to conservative partitioning reflects that the aggressive partitioning is not satisfactory. A switch in the opposite direction is tentative since it is only known that the conservative partitioning *is* satisfactory so the aggressive partitioning *may* also be satisfactory. Therefore the first step after a switch in this direction is monitored.

The algorithm was developed and presented for the decoupled implicit Euler formula. The modifications, beyond the obvious, to accommodate the decoupled BDF2 are small. The Euler formula is replaced by the decoupled BDF2 (2.11) in mode 3, and the first order predictor is replaced by the second order predictor (4.2) except the first integration step which is still taken by the Euler formula and the second step which uses decoupled BDF2, mode 2.

The principal local error is estimated using (4.7), and a normalization by β of the local error estimate is introduced [6, Section III.2.], $\varepsilon_{est} = \|(Y_n - Y_n^{p2})C_3/\bar{C}_3/\beta\|$. The step size selection scheme is modified slightly so that averaging is only used during increasing step sizes:

$$\rho = (\varepsilon_{tol}/\varepsilon_{est})^{1/3}; \quad \textbf{if } \rho > 1 \textbf{ then } h_{n+1} = \frac{h_n}{2}(1+\rho) \textbf{ else } h_{n+1} = h_n\rho.$$

The step size averaging is useful for increasing step sizes to reduce the risk of instability while it prevents a very rapid reduction in step size at the transients.

6. Examples: Chemical reaction kinetics.

The example problem is the mathematical model of the chemical reactions included in a three-dimensional transport-chemistry model of air pollution. The air pollution model is a system of partial differential equations where each equation models transport, deposition, emission, and chemical reactions of a pollutant. By the use of operator splitting, a number of sub-models are obtained including the following system of 32 nonlinear ODEs:

$$Y_i' = P_i(t, Y) - L_i(t, Y)Y_i, \quad i = 1, 2, \ldots, 32.$$

The nonlinearities are mainly products, i.e., P_i and L_i are typically sums of terms of the form, $c_{ilm}(t)Y_lY_m$ and $d_{il}(t)Y_l$, respectively, for $l, m \neq i$.

The chemistry model is replicated for each node of the spatial discretization of the transport part. The numerical solution of a system of 32 ODEs is not very challenging as such, but the replication results in hundreds of thousands or millions of equations, and a very efficient numerical solution is crucial. The problem and a selection of solution techniques employed so far are described in [9] and [10].

The system of ODEs is very stiff, with the real part of the eigenvalues of the Jacobian along the solution ranging from 0 to $-8 \cdot 10^4$ and step sizes in the range 100 to 1000. Therefore, implicit integration schemes are required, and the resulting nonlinear algebraic problem is the main computational task involved in advancing the numerical solution one time step. A method based on partitioning the system called the Euler backward iterative method is described in [9]. It can be characterized as a discretization by the implicit Euler formula with block Gauss–Seidel iteration for the solution of the algebraic equations of the discretization.

An ideal partitioning would involve subsystems of size one, i.e., $s_i = 1$ for all i (cf. (2.2)). For this chemical reaction kinetics problem, it would be particularly advantageous since Y_i is only included in $L_i(t, Y)$ in very few equations and never in $P_i(t, Y)$ (by definition). Therefore $L_i(t, Y)Y_i$ is in general linear in Y_i, and the solution of a scalar equation by an implicit integration formula can be performed without iterations.

A partitioning into all scalar equations is not viable for this problem. However, the paper [9] identifies a total of 12 equations which should be solved in blocks of 4, 4, 2, and 2 equations. With the numbering of the chemical species used in Table B.1 in [9], the block Gauss–Seidel iteration proceeds as follows, where the parentheses denote the blocks of equations {(1, 2, 3, 12), (4, 5, 19, 21), 6, 7, (8, 9), 10, 14, (15, 16), remaining scalar equations}. The partitioning in [9] specified above is used as the basis of the partitioning used for the results in [11].

In [9] the Euler backward iterative formula is relaxed until convergence to obtain the equivalent of the classical implicit Euler formula. The results in [11] are obtained from the decoupled implicit Euler formula mode 2 with one relaxation. Timing results in [11] show good efficiency for this approach.

The aggressive partitioning and ordering used in this example for both decoupled implicit Euler and BDF2 is described by {12, (4, 5), 20 scalar equations, (1, 2, 15, 16), remaining scalar equations}. The subsystems of two and four equations are enclosed in parentheses, and the rest of the equations are being treated as scalar equations.

The conservative partitioning and ordering is specified by {12, (1, 2, 15, 16, 4, 5, 8, 10, 29, 30), remaining scalar equations}. Except for the block of 10 equations, the partitioning is into scalar equations.

One partitioning is considered more aggressive than another one if it has fewer equations appearing in blocks and/or the blocks are smaller. Concerning the two partitionings presented here, it is obvious which partitioning is the more aggressive since $\{(4, 5), (1, 2, 15, 16)\} \subset \{(1, 2, 15, 16, 4, 5, 8, 10, 29, 30)\}$. The parentheses are only retained to facilitate reference to the partitioning and ordering specifications.

The aggressive partitioning is clearly more aggressive than the partitioning presented in [9] since $\{(4, 5), (1, 2, 15, 16)\} \subset \{(1, 2, 3, 12), (4, 5, 19, 21), (8, 9), (15, 16)\}$, while the relation to the conservative partitioning is not obvious since one has the larger block while the other has more equations appearing in smaller blocks.

The partitioning used in [9] is presumably based on knowledge of the chemical reactions, while the partitionings used here are obtained with a semiautomatic method described in paper [12]. None of these partitionings are monotonically max-norm stable although they come close, but despite this fact, no instability problems are encountered. This should not be too surprising since the monotonic max-norm stability is a sufficient but not a necessary condition for the stability of the decoupled implicit Euler formula.

The example used in this paper differs slightly from the example in [9], but the differences appear in the equations being treated as scalar in all the considered partitionings.

6.1. Decoupled implicit Euler.

Figure 6.1 shows the errors obtained using the classical implicit Euler formula (solid line) and the decoupled implicit Euler formula (dash-dot line) implemented as described in section 5. The two different partitionings mentioned above have been used adaptively. The step size is chosen by the

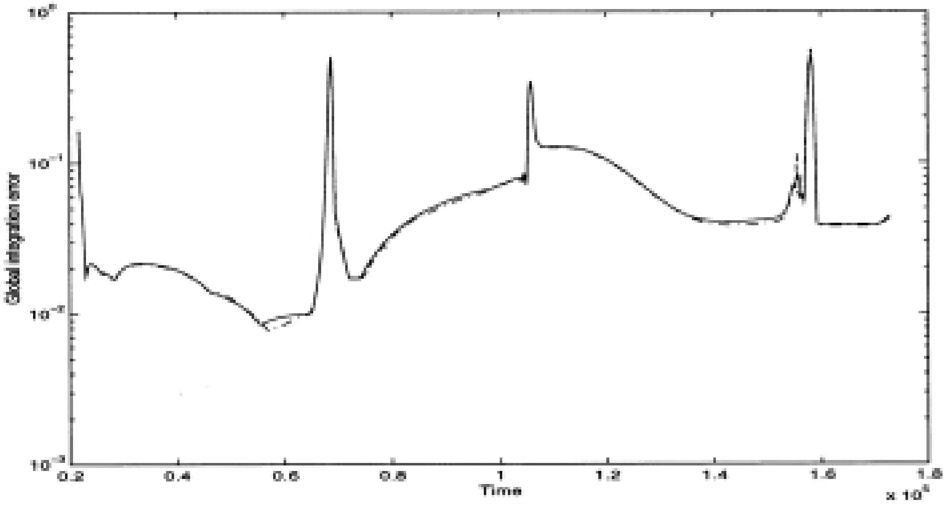

FIG. 6.1. *Integration error for the classical implicit Euler (solid line) and decoupled implicit Euler (dash-dot line).*

decoupled implicit Euler algorithm to obtain a local error estimate of 10^{-3} or to a minimum step size of 90, and the classical implicit Euler formula is applied with the same step size selection. The discrepancy between the errors is seen to be insignificant.

A reference solution is computed using a variable step size variable order (maximum order = 6) implementation of the backward differentiation formulas [13] with a bound on the relative local error estimate of 10^{-6}. The errors presented in the figures are the maximum relative deviations from the reference solution measured componentwise (the values of the components vary widely in magnitude).

The time axis is in seconds, and the initial time corresponds to 6 a.m. The model includes the influence of the sun on some of the chemical reactions, and this leads to very distinct transients in the solution at sunrise and sunset. The minimum integration time step of 90 seconds is too large a step to integrate the transients accurately, and large spikes in the global integration error are seen around 7 p.m., 5 a.m. (t=105,000) the next day, and 7 p.m. (t=155,000).

The transient behavior at sunrise and sunset can to some extent be considered a modelling artifact. Since the large local contribution to the global error does not influence the global error at large, it is essentially ignored by introducing the minimum time step. The observed behavior of the global error is not uncommon for stiff systems of ODEs.

Figure 6.2 shows the estimated principal local error. The step size is adjusted to keep the estimate at 10^{-3}, and the algorithm is quite successful except at the transients where the minimum step size of 90 is used.

Figure 6.3 shows the resulting step size selection (upper graph) and the selection of partitioning (lower graph). The low value indicates the aggressive partitioning, and the high value indicates the conservative partitioning. The values on the ordinate axis only pertain to the step size. The total number of steps is 737, and only 182 (25%) steps are using the conservative partitioning.

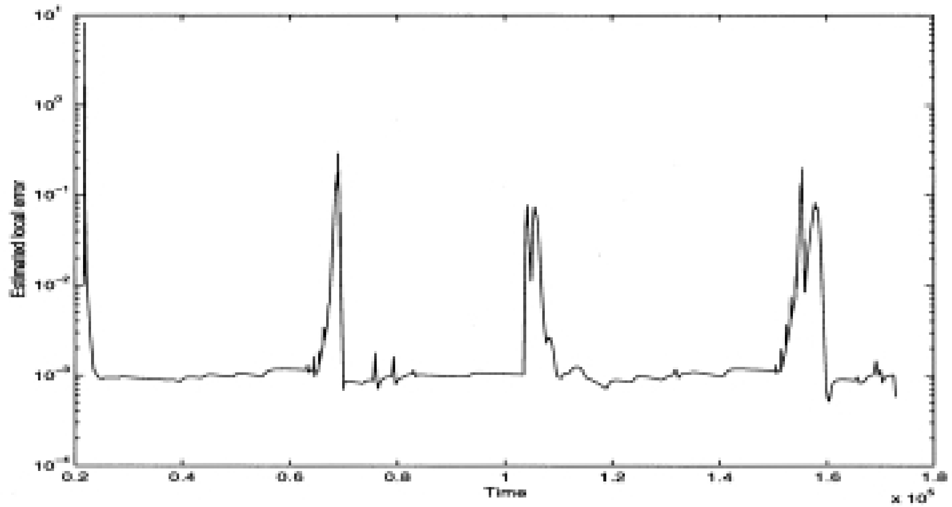

Fig. 6.2. *Estimated principal local error for decoupled implicit Euler.*

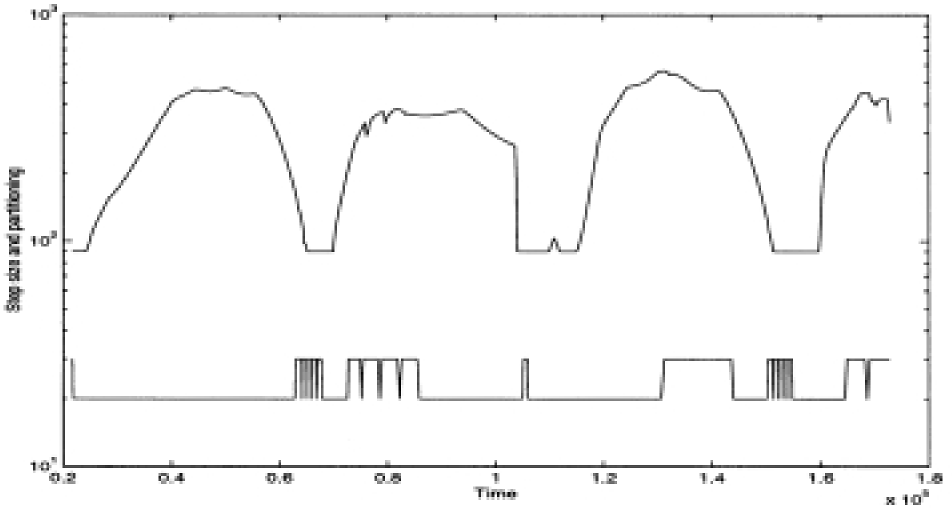

Fig. 6.3. *Integration step size selected by the decoupled implicit Euler formula (upper graph) and selection of partitioning (lower graph).*

During integration using the conservative partitioning, single steps that use the aggressive partitioning can be observed. At the first step after a switch to the aggressive partitioning, the quality is monitored, and if it is not satisfactory, the *Monitor partitioning* algorithm immediately returns to the conservative partitioning.

This example demonstrates the application of an implementation of the decoupled implicit Euler formula in solving a nontrivial practical problem. The computational cost is substantially less than that for the classical implicit Euler formula, and there is no trace of instability in the solution computed by the decoupled implementation.

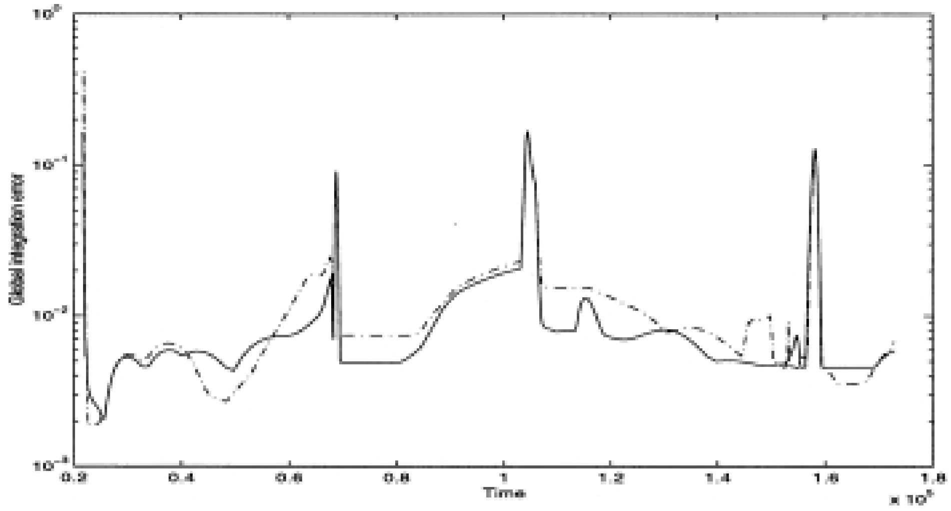

FIG. 6.4. *Integration error for the classical BDF2 (solid line) and decoupled BDF2 (dash-dot line).*

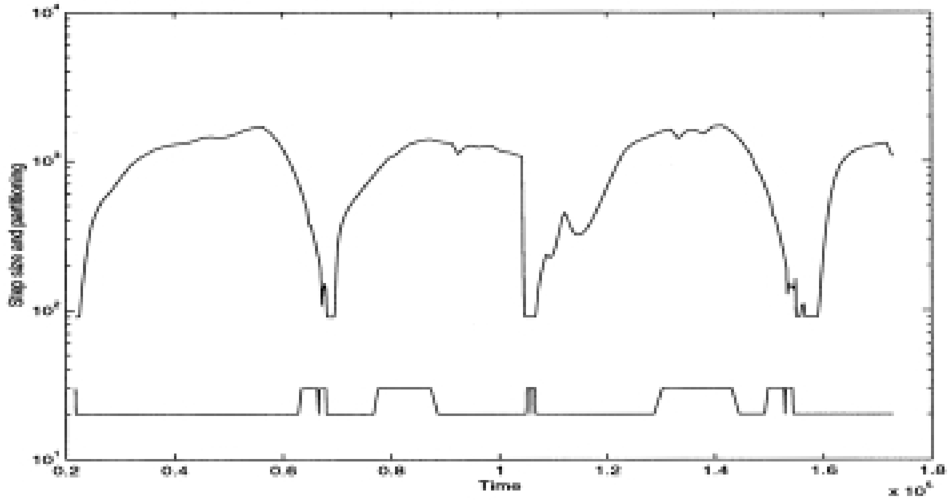

FIG. 6.5. *Integration step size selected by the decoupled BDF2 (upper graph) and selection of partitioning (lower graph).*

6.2. Decoupled BDF2. The numerical integration of the previous section is repeated using the decoupled BDF2. The step size is controlled to keep the estimated local normalized error ε_{est} at 10^{-3} as before. The resulting global integration error is shown in Figure 6.4 (dash-dot line) together with the corresponding error for the classical BDF2 using the same step size selection (solid curve). Some deviation is noticed, but the error of the decoupled BDF2 is comparable to the error of the classical BDF2 formula. Comparing to Figure 6.1, it is seen that the global error of the decoupled BDF2 formula is significantly smaller than the global error of the decoupled implicit Euler formula. The difference originates mainly from the transients, including the initial transient, where the minimum step size of 90 seconds is used.

Finally, Figure 6.5 shows step size and a partitioning selection similar to Figure 6.3. The maximum step size of the decoupled BDF2 is greater than 1700 which is three times the maximum step size of the Euler formula. The necessary number of integration steps is 311 which is 42% of the number of steps needed by the Euler formula. The conservative partitioning is only used for 73 steps out of 311 (23%). There is a substantial pay-off to using the decoupled BDF2 instead of the decoupled Euler formula, since the amount of work per step is essentially the same for the two decoupled formulas.

The performance of the decoupled BDF2 algorithm in mode 3 is very convincing, and there is no trace of instability, although the use of the second order polynomial predictor for computing $\tilde{Y}_n$ is somewhat risky from a stability point of view.

Acknowledgements. The comments by the editor and the anonymous referees have helped choosing asymptotically correct error estimates and improving the presentation.

REFERENCES

[1] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Trans. Computer-Aided Design Integrated Circuits Systems, 1 (1982), pp. 131–145.

[2] S. Skelboe, *Methods for parallel integration of stiff systems of ODEs*, BIT, 32 (1992), pp. 689–701.

[3] J. Sand and S. Skelboe, *Stability of backward Euler multirate methods and convergence of waveform relaxation*, BIT, 32 (1992), pp. 350–366.

[4] L. M. Adams and H. F. Jordan, *Is SOR color-blind?*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 490–506.

[5] J. White and F. Odeh, *A connection between the convergence properties of waveform relaxation and the A-stability of multirate integration methods*, in Proceedings of the Seventh International Conference on the Numerical Analysis of Semiconductor Devices and Integrated Circuits, Copper Mountain, CO, April 8–12, 1991, J. J. H. Miller, ed., Front Range Press, Boulder, CO, 1991, pp. 73–76.

[6] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I, Springer-Verlag, Berlin, 1987.

[7] S. Skelboe, *The control of order and steplength for backward differentiation methods*, BIT, 17 (1977), pp. 91–107.

[8] C. W. Gear, *Estimation of errors and derivatives in ordinary differential equations*, in Information Processing 74, North-Holland, Amsterdam, 1974, pp. 447–451.

[9] O. Hertel, R. Berkowicz, J. Christensen, and Ø. Hov, *Test of two numerical schemes for use in atmospheric transport-chemistry models*, Atmospheric Environment, 27A (1993), pp. 2591–2611.

[10] M. W. Gery, G. Z. Whitten, J. P. Killus, and M. C. Dodge, *A photochemical kinetics mechanism for urban and regional computer modelling*, J. Geophys. Res., 94 (1989), pp. 12925–12956.

[11] S. Skelboe and Z. Zlatev, *Exploiting the natural partitioning in the numerical solution of ODE systems arising in atmospheric chemistry*, in Proceedings of the First International Workshop (WNNA-96), Rousse, Bulgaria, June 24–26, 1996, Lecture Notes in Comput. Sci. 1196, L. Vulkov, J. Wasniewski, and P. Yalamov, eds., Springer-Verlag, Berlin, Germany, 1997, pp. 458–465.

[12] S. Skelboe, *Partitioning Techniques and Stability of Decoupled Integration Formulas for ODEs*, in preparation.

[13] S. Skelboe, *INTGR for the Integration of Stiff Systems of Ordinary Differential Equations*, Report IT 9, Institute of Circuit Theory and Telecommunication, Technical University of Denmark, 1977.