Datalogisk Institut
KØBENHAVNS UNIVERSITET

DIKU

Partitioning Techniques for ODEs for Decoupled Implicit Intergration Formulas

Stig Skelboe

Technical Report no. 05/04
ISSN: 0107-8283

Dept. of Computer Science
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Partitioning Techniques for ODEs for Decoupled Implicit Integration Formulas

Stig Skelboe [*]

January 28, 2004

Abstract

Many stiff systems of ordinary differential equations (ODEs) modeling practical problems can be partitioned into loosely coupled subsystems. In this paper the objective of the partitioning is to permit the numerical integration of one time step to be performed as the solution of a sequence of small subproblems. This reduces the computational complexity compared to solving one large system and permits efficient parallel execution under appropriate conditions. The subsystems are integrated using methods based on low order backward differentiation formulas.

This paper presents techniques for the partitioning of systems of ODEs based on a classical graph algorithm. The complexity of a partitioned discretization is evaluated using operations count, and the paper presents a selection of techniques for the efficient evaluation of the error introduced by the partitioning.

The feasibility of the approach is demonstrated by an experimental integration algorithm which, along the solution, adaptively partitions a system of ODEs originating from chemical reaction kinetics. The computational savings are reported to be substantial.

1 Introduction

The numerical solution of a stiff system of S ordinary differential equations (ODEs) typically has computational complexity $\mathcal{O}(S^3)$ because of the linear algebra. Besides, it is difficult to solve stiff systems efficiently on a parallel computer when the only option is to parallelize the linear algebra.

[*] Technical report 05/04, Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark, e-mail: stig@diku.dk

Waveform relaxation [1] was originally developed to exploit the structure of digital circuits to reduce the amount of linear and nonlinear algebra involved in numerical integration. Later it was realized that waveform relaxation was very well suited as the basis for parallel numerical integration algorithms.

In [2] an approach similar to waveform relaxation is proposed but without the relaxation part. This reduces the amount of computation but requires a more precise partitioning into subsystems. Besides, the computational granularity is finer, which means that parallel speed-up is expected to be smaller than for waveform relaxation. The paper [2] deals with the absolute stability and global error expansion of decoupled backward differentiation formulas, especially the implicit Euler formula. The formulas are demonstrated on a toy example.

In [3] the decoupled implicit Euler and second order backward differentiation formulas (BDF2) are developed for practical applications, and expressions for the local truncation errors are presented. The formulas are demonstrated on a real problem, the same as the one in this paper section 6. The example in [3] relies on two a priori partitionings where an algorithm adaptively during the integration selects the more efficient (i.e. smallest subsystems) which is sufficiently accurate.

This paper presents low-cost formulas for the evaluation of the accuracy of a partitioning and a graph based technique to perform the partitioning. At present it is not clear that it is possible to design an efficient *general* purpose integration package based on adaptive partitioning and decoupled integration formulas. Therefore this paper presents a selection of elements which may be of use in the design of a *special* purpose integration program for a well defined class of problems. The feasibility of constructing such a program is illustrated by the algorithm in section 5.4 and the integration of the example problem in section 6.

The motivation in this paper for being interested in the partitioning of a system of ODEs, namely the potential reduction in operations count, is given in section 2 together with an introduction of the decoupled implicit Euler formula.

Section 3 presents a rather general analysis of the error of the decoupled implicit Euler formula. In the analysis, the influence of the partitioning becomes clear, and the analysis gives a qualitative understanding of the properties of a good partitioning. However, the analysis gives fairly pessimistic bounds and it is expensive and difficult to perform in general.

In practical applications, an analysis of the linearized problem is usually to be preferred and certainly when the partitioning is performed adaptively. Section 4.1.1 discusses the splitting error which is independent of the discretization. In section 4.1.2 techniques are presented for the evaluating of the difference between the matrix resolvent M_E for the classical Euler and the corresponding matrix M_D for the decoupled implicit Euler formula. The matrix difference is evaluated for typical linearizations of the problem, and it is independent of the actual solution. This matrix difference was used in [3] for the a priori selection of partitionings.

When a system is partitioned adaptively, the linearization is performed at the current solution point, and it is obvious to evaluate the error of the *solution* resulting from the partitioning in stead of the matrix difference. This approach permits a very accurate evaluation of the partitioning error, and section 4.2 with subsections present various techniques.

In particular adaptive partitioning requires very efficient techniques for the evaluation of a partitioning, and both section 4.1 and 4.2 present approximations to reduce the computational cost of the evaluation as far as possible.

Section 5.1 presents the theoretical basis for a partitioning approach and then in section 5.2 the reordering algorithm used to obtain a partitioned system. These are put together in section 5.3 to make up the framework of a partitioning algorithm. Section 5.4 presents the specification of an adaptive partitioning algorithm used to solve the example problem in section 6.

This example problem was also solved in [3], and it is clear that the adaptive partitioning algorithm is able to obtain more aggressive partitionings than the approach applied in [3]. The performance of the decoupled Euler formula is illustrated by graphs showing the variation of key values.

2 Partitioned systems of ODEs and decoupled discretization formulas

Define a system of ODEs,

$$Y' = F(t, Y),\ Y(t_0) = Y_0 \text{ and } t \geq t_0 \tag{1}$$

where $Y : R \rightarrow R^S$, $F : R \times R^S \rightarrow R^S$, 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 implicit integration formulas be exemplified by the backward differentiation formulas (BDF),

$$Y_n = \sum_{j=1}^{k} \alpha_j Y_{n-j} + h_n \beta_0 F(t_n, Y_n) \tag{2}$$

The implicit formulas require Newton-type iteration to advance the solution one time-step, e.g.

$$Y_n^{[m+1]} = Y_n^{[m]} - \left(I - h_n \beta_0 \frac{\partial F}{\partial Y} \right)^{-1} \left(Y_n^{[m]} - \sum_{j=1}^{k} \alpha_j Y_{n-1} - h_n \beta_0 F(t_n, Y_n^{[m]}) \right) \tag{3}$$

The computational complexity of F and $h_n \beta_0 \partial F / \partial Y$ in terms of floating point computations per function or Jacobian evaluation is in the following assumed to be $\eta_F S^2$ and $\eta_J S^2$, respectively. The complexity is of main interest when it is valid for classes of problems parametrized by S. An obvious example is the system of ODEs resulting from applying the method of lines to PDEs. The term S^2 is probably a worst case value, and S^p where $1 \leq p \leq 2$, may be more realistic, corresponding to a sparse Jacobian.

The total complexity of the Newton iteration (3) with n_{it} iterations is then,

$$C_{BDF} = \eta_J S^2 + \frac{2}{3} S^3 - \frac{1}{2} S^2 - \frac{1}{6} S + (2k-1)S + n_{it} \left[\eta_F S^2 + 4S + 2S^2 \right] \tag{4}$$

The complexity expression assumes a pseudo-Newton scheme with just one Jacobian evaluation and LU-factorization for each solution step. The complexity of the LU-factorization and corresponding solution stage applies to a full matrix algorithm. The complexity of a sparse LU-factorization and solution are dominated by terms like S^{p_f}, $2 \leq p_f \leq 3$ and S^{p_s}, $1 \leq p_s \leq 2$, respectively.

When big systems of stiff ODEs are solved, it is important to be able to perform the linear and nonlinear algebra as efficiently as possible. The techniques described in this paper aim at reducing the computational complexity and furthermore permit efficient use of parallel computation. However, the focus is on reducing complexity.

Let the original problem (1) be partitioned as follows:

$$\begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_q' \end{pmatrix} = \begin{pmatrix} f_1(t,Y) \\ f_2(t,Y) \\ \vdots \\ f_q(t,Y) \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{pmatrix}, \quad Y(t_0) = \begin{pmatrix} y_{1,0} \\ y_{2,0} \\ \vdots \\ y_{q,0} \end{pmatrix} \tag{5}$$

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

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]:

$$y_{r,n} = y_{r,n-1} + h_n f_r(t_n, \tilde{y}_{1,n}, \ldots, \tilde{y}_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \ldots, \tilde{y}_{q,n}) \qquad (6)$$

where $n = 1, 2, \ldots,$ $t_n = t_0 + \sum_{j=1}^{n} h_j$, 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 (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 (6), for different r-values, is solved independently and in parallel.

Theorems 4 and 5 in [4] assure the stability of the discretization by (6) of a monotonically max-norm stable problem and the convergence of waveform relaxation, and as a special case, the convergence of the point-wise relaxation as defined in (2.9) in [3]. The stability condition poses no restrictions on the choice of the step sizes h_n, and even multi-rate solution is covered. The convex representation of $\tilde{y}_{r,n}$ is necessary in the stability theorem of the decoupled implicit Euler method. A convex combination of previous solution values would typically be a zero-order interpolation, $\tilde{y}_{r,n} = y_{r,n-1}$.

It is straightforward to extend the principle of the decoupled implicit Euler formula to higher order backward differentiation formulas (2). The decoupled BDF2 is analyzed in [3] and even higher order decoupled BDF might be considered. However, only the stability properties of the decoupled implicit Euler formula are well-studied [4].

The sequential solution of (6) for $r = 1, 2, \ldots, q$ on a single processor will, in general, be computationally cheaper than solving the complete system $Y_n = Y_{n-1} + h_n F(t_n, Y_n)$. 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.

The computational complexity of subsystem r of the decoupled implicit Euler formula is assumed to be,

$$C_{DEr} = \eta_J s_r^2 + \frac{2}{3} s_r^3 - \frac{1}{2} s_r^2 + \eta_F s_r S + n_{it} \left[\eta_F s_r^2 + 4 s_r + 2 s_r^2 \right]$$

The assumptions are the same as in (4). Besides, it is assumed that the number of iterations for each subsystem, n_{it}, is the same for all subsystems and the same as for the classical Euler formula. However, this needs not be the case. In the example in section 6, most of the the subsystems are actually scalar problems, and most of these turn out to be linear in the unknown thus reducing the iteration count to 1.

The computational complexity of f_r is somewhere between $\eta_F s_r^2$ and $\eta_F s_r S$. In the model in (7), it is assumed that after an initialization of complexity $\eta_F s_r S$ before the iteration loop, f_r can be computed at the cost of $\eta_F s_r^2$.

The computational complexity of a decoupled k-step BDF analogue to (4) is as follows assuming that $s_i = S/q$,

$$C_{DBDF} = \eta_J \frac{1}{q} S^2 + \frac{2}{3}\frac{1}{q^2} S^3 - \frac{1}{2}\frac{1}{q} S^2 + \eta_F S^2 + (2k-1)S + n_{it}\left[\eta_F \frac{1}{q} S^2 + 4S + 2\frac{1}{q} S^2\right] \tag{7}$$

The expected speed-up compared with the classical BDF is somewhere from q^2, when the computation is dominated by linear algebra, to $q/[1+q/(1+n_{it})]$ when the nonlinear algebra dominates and $\eta_F \approx \eta_J$.

The objective of this paper is thus to devise partitioning strategies and algorithms that result in partitioned systems (5) with as small subsystems as possible without compromising accuracy and stability of the decoupled Euler discretization (6) relative to the classical Euler discretization.

The decoupled implicit Euler formula is related to the implicit-explicit (IMEX) multistep methods primarily considered for the solution of partial differential equations. The IMEX methods assume the following structure of the system of ODEs,

$$Y' = F_1(t, Y) + F_2(t, Y) \tag{8}$$

where $Y' = F_1(t, Y)$ is a stiff problem and $Y' = F_2(t, Y)$ is a non-stiff problem. The problem is then solved using a splitting method based on a stiff and a non-stiff solver.

3 Partitioning Accuracy

The accuracy and computational efficiency of the decoupled implicit Euler formula (6) depends critically on the partitioning of the original problem (1) into (5). The partitioning algorithm described in section 5 is iterative and requires an efficient method for evaluating the accuracy of a proposed partitioning.

Along with the integration using the decoupled implicit Euler formula, the accuracy should be monitored. The accuracy is likely to change because of nonlinearities of the problem and also for a linear problem with the change of the solution.

A partitioning is supposed to be applicable for many time steps, maybe for all of the solution while the accuracy must be monitored continually. Therefore computationally more expensive – and accurate methods – may be chosen for the evaluation of the accuracy of the partitioning algorithm while less accurate and cheaper methods should be chosen for the monitoring along the solution. First a general nonlinear analysis aimed at providing insight will be presented.

Consider the partitioning of $F(t, Y)$ as specified in (5). The partitioned system and corresponding discretizations can be characterized by the $q \times q$ matrix $A(t, U, V)$ defined as follows, where $U, V \in \Omega$ and $\Omega \subseteq R^S$ is a convex neighborhood of Y.

$$\begin{aligned} a_{rr}(t, U, V) &= \mu(B_{rr}(t, U, V)) \\ a_{rj}(t, U, V) &= \|B_{rj}(t, U, V)\|,\ r \neq j \end{aligned}$$

Here μ denotes the logarithmic norm which is derived from the matrix norm which is again derived from the vector norm used in this section. The matrices B_{rj} of dimension $s_r \times s_j$ are defined as follows, assuming sufficient differentiability:

$$(B_{r1}(t, U, V), B_{r2}(t, U, V), \ldots, B_{rq}(t, U, V)) = \int_0^1 \partial f_r(t, \phi U + (1-\phi)V)/\partial Y d\phi$$

Then the following inequality holds (Theorem 3 [4]):

$$\|u_r - v_r + \lambda\left[f_r(t, U) - f_r(t, V)\right]\| \geq \|u_r - v_r\| + \lambda \sum_{j=1}^{q} a_{rj}(t, U, V)\|u_j - v_j\|$$

for $\lambda \leq 0$ and $U, V \in \Omega$.

The partitioned system (5) is said to be monotonically max-norm stable [4] if $\mu_\infty(A(t, U, V)) \leq 0$ for $U, V \in \Omega$. Theorems 4 and 5 in [4] assure the stability of the discretization of a monotonically max-norm stable partitioned system by the decoupled implicit Euler formula (6).

Consider the decoupled implicit Euler formula $y_{r,n} - y_r(t_{n-1}) - h_n f_r(t_n, \tilde{Y}_n^r) = 0$, where $\tilde{Y}_n^r = (\tilde{y}_{1,n}, \ldots, \tilde{y}_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \ldots, \tilde{y}_{q,n})$, and the local truncation error for subsystem r,

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

Introduce the simplified notation $\tilde{a}^n_{rj} = a_{rj}(t_n, Y(t_n), \tilde{Y}^r_n)$. Then subtraction leads to

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

Define a convex neighborhood of $Y(t_n)$ called Ω_{t_n}. Assume that $[1 - h_n\tilde{a}^n_{rr}] > 0$ for $\tilde{Y}^r_n \in \Omega_{t_n}$ for all r. Then a bound for the local error $y_r(t_n) - y_{r,n}$ $(r = 1, 2, \ldots, q)$ of the decoupled implicit Euler formula for $\tilde{Y}^r_n \in \Omega_{t_n}$ can be expressed as follows (cf. Lemma 2.2, section III.2. in [5]),

$$\|y_r(t_n) - y_{r,n}\| \leq [1 - h_n\tilde{a}^n_{rr}]^{-1} \left(\|\mathcal{L}[Y(t_n); h_n]_r\| + h_n \sum_{j\neq r} \tilde{a}^n_{rj}\|y_j(t_n) - \tilde{y}_{j,n}\| \right) \tag{9}$$

The local truncation error also appears in the error of the classical implicit Euler formula, but the decoupling introduces a new error specified in the summation term in (9).

It is clear from (9) that a good partitioning, from an accuracy point of view, has numerically large negative values in the diagonal of $A(t, U, V)$ for $U, V \in \Omega_{t_n}$ and numerically small off-diagonal values.

The definition of $\tilde{Y}^r_n$ corresponds to a Jacobi type of organization of the decoupled Euler formula, and this organization permits parallel execution of each subsystem. When this is not relevant, a Gauss-Seidel organization will in general result in better accuracy. Define $\bar{Y}^r_n$ as follows,

$$\bar{Y}^r_n = (y_{1,n}, \ldots, y_{r-1,n}, y_{r,n}, \tilde{y}_{r+1,n}, \ldots, \tilde{y}_{q,n})$$

and the corresponding $\bar{a}^n_{rj} = a_{rj}(t_n, Y(t_n), \bar{Y}^r_n)$. Assuming that $\bar{Y}^r_n \in \Omega_{t_n}$ for all r and $[1 - h_n\bar{a}^n_{rr}] > 0$ for $\bar{Y}^r_n \in \Omega_{t_n}$, then the error bound for the Gauss-Seidel organization analogous of (9) is,

$$\begin{aligned} \|y_r(t_n) - y_{r,n}\| \leq{}& [1 - h_n\bar{a}^n_{rr}]^{-1} \Big(\|\mathcal{L}[Y(t_n); h_n]_r\| \qquad (10) \\ & + h_n \sum_{j<r} \bar{a}^n_{rj}\|y_j(t_n) - y_{j,n}\| + h_n \sum_{j>r} \bar{a}^n_{rj}\|y_j(t_n) - \tilde{y}_{j,n}\| \Big) \end{aligned}$$

Assuming that $\|y_j(t_n) - y_{j,n}\| < \|y_j(t_n) - \tilde{y}_{j,n}\|$ and $\tilde{a}^n_{rj} \approx \bar{a}^n_{rj}$, the bound of the local error of the Gauss-Seidel organization (11) is in general smaller than the bound of the local error of the Jacobi organization (9).

Furthermore a reordering of the partitioned system corresponding to a symmetric row-column reordering of A can put the numerically largest off-diagonal elements of A into the lower triangular part where the impact on the local error is smaller due to the assumption $\|y_j(t_n) - y_{j,n}\| < \|y_j(t_n) - \tilde{y}_{j,n}\|$.

The analysis presented in this section is rather general, and it provides both qualitative and quantitative understanding of the error of the decoupled implicit Euler formula resulting from the partitioning.

When used for evaluating the partitioning of a problem, two points should be noted. First, the local truncation error (or an estimate) is needed. Second, the use of norms may lead to error bounds that are unduly pessimistic. In [4] the analysis is based on different norms for different subsystems. By choosing the norms appropriately, some of the pessimism can be removed at the cost of a more complicated analysis.

The next section presents analysis of the partitioning based on a linearization of the problem. The linear analysis can be used as an alternative or a supplement to the analysis of this section.

Example 1

Consider a linear problem $Y' = BY$ of dimension $S = 4$. Let B be partitioned into four 2×2 blocks ($s_i = 2$),

$$B_{11} = \begin{pmatrix} -2 & 1 \\ 0 & -10 \end{pmatrix}, \; B_{12} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; B_{21} = \begin{pmatrix} 0 & 10 \\ 1 & 0 \end{pmatrix}, \; B_{22} = \begin{pmatrix} -2 & 0 \\ 10 & -20 \end{pmatrix}$$

With $Y(0) = (1, 1, 1, 1)^T$, consider the point of the smooth solution,

$$Y(1) = (4.4589_{10}{-1},\ 8.3613_{10}{-2},\ 7.6067_{10}{-1},\ 4.2157_{10}{-1})^T$$

Let this point be the initial point, $Y(1) = (y_{1,0},\ y_{2,0})^T$, for a numerical integration using the decoupled implicit Euler formula with $\tilde{y}_{j,1} = y_{j,0}$, Jacobi organization and $h = 0.1$.

Using the infinity norm throughout this example, we have $\mu(B_{11}) = -1$, $\|B_{12}\| = 1$, $\|B_{21}\| = 10$, and $\mu(B_{22}) = -2$. Notice that the infinity norm does not establish monotonical max-norm stability of this partitioned system. The error of one step is bounded using (9), and the required exact solution is easily computed using e.g. Matlab.

$$\|y_1(1.1) - y_{1,1}\| \le \frac{1}{1+0.1}(1.5189_{10}{-3} + 0.1 \times 1 \times 3.7103_{10}{-2}) = 4.7538_{10}{-3}$$

$$\|y_2(1.1) - y_{2,1}\| \le \frac{1}{1+0.2}(2.9041_{10}{-3} + 0.1 \times 10 \times 6.9530_{10}{-2}) = 5.7028_{10}{-2}$$

With the Gauss-Seidel organization, $y_{2,1}$ is different, and the error bound is given by (11),

$$\|y_2(1.1) - y_{2,1}\| \leq \frac{1}{1+0.2}(2.9041_{10}{-3} + 0.1 \times 10 \times 8.4292_{10}{-3}) = 9.4444_{10}{-3}$$

The results are summarized for easy comparison in Table 1.

	Jacobi organization		Gauss-Seidel organization	
	error	bound (9)	error	bound (11)
$\|y_1(1.1) - y_{1,1}\|$	$4.5723_{10}{-3}$	$4.7538_{10}{-3}$	$4.5723_{10}{-3}$	$4.7538_{10}{-3}$
$\|y_2(1.1) - y_{2,1}\|$	$8.4292_{10}{-3}$	$5.7028_{10}{-2}$	$5.2852_{10}{-3}$	$9.4444_{10}{-3}$

Table 1: Summary of first example problem results

In this example, the bound for $\|y_1(1.1) - y_{1,1}\|$ is fairly tight while it is somewhat looser for $\|y_2(1.1) - y_{2,1}\|$. However, both the bound and the actual value of $\|y_2(1.1) - y_{2,1}\|$ is smaller for the Gauss-Seidel organization than for the Jacobi organization as expected.

If the example is modified slightly, by transposing B_{21}, the matrix norms are unchanged. Bounds and errors for the modified example are summarized in Table 2.

	Jacobi organization		Gauss-Seidel organization	
	error	bound (9)	error	bound (11)
$\|y_1(1.1) - y_{1,1}\|$	$5.2092_{10}{-3}$	$5.9552_{10}{-3}$	$5.2092_{10}{-3}$	$5.9552_{10}{-3}$
$\|y_2(1.1) - y_{2,1}\|$	$1.6191_{10}{-2}$	$3.4595_{10}{-2}$	$3.3755_{10}{-3}$	$1.5687_{10}{-2}$

Table 2: Summary of second example problem results

Comparing the bounds and the errors for the Gauss-Seidel organization for the two versions of the example, they change in opposite direction. This is a result of using norms in stead of more detailed measures.

4 Linear Analysis

The purpose of the analysis in this section is to provide the theoretical understanding and foundation for the partitioning technique to be presented in the next section. Besides, the analysis provides tools for the evaluation of a partitioning.

The general nonlinear system of ODEs (1), transformed into an autonomous system, can be linearized around a point (t_0, Y_0) leading to a system of the type,

$$Y' = BY + V, \;\; Y(t_0) = Y_0 \tag{11}$$

where Y is assumed to include t and the system (1) is augmented with the equation $t' = 1$.

Some of the analysis to follow is simpler when applied to the homogeneous system $Y' = BY$, and for some analysis is does not make any difference whether it is applied to the homogeneous problem or to (11).

The partitioned linear homogeneous problem corresponding to (5) is as follows,

$$\begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_q' \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} & ... & B_{1q} \\ B_{21} & B_{22} & ... & B_{2q} \\ ... & ... & ... & ... \\ B_{q1} & B_{q2} & ... & B_{qq} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{pmatrix}$$

Define $D = \text{diag}(B_{11}, B_{22}, ..., B_{qq})$ and $E = B - D$. With this definition of D and E the decoupled implicit Euler formula,

$$Y_n^{[1]} = Y_{n-1} + h(DY_n^{[1]} + EY_{n-1}) \tag{12}$$

corresponds to the Jacobi type organization, mode 1 [3]. The mode 1 decoupled implicit Euler formula (where $\tilde{Y}_n = Y_{n-1}$) leads to simple formulas for the evaluation and comparison of partitionings. However, in practice a higher mode and/or a decoupled BDF2 formula will be used. Therefore the estimates of the decoupling error from the formulas presented in this section may be pessimistic.

Alternatively D can be defined as the lower block-triangular part (including the block-diagonal) of B. Then (12) corresponds to the Gauss-Seidel organization, mode 1 in [3]. Solving for $Y_n^{[1]}$ from (12) leads to $Y_n^{[1]} = M_D Y_{n-1}$ where

$$M_D = (I - hD)^{-1}(I + hE)$$

For a stiff system (11) it is assumed that the following relations are fulfilled during the integration of the smooth solution,

$$\rho(M_D) \leq 1, \;\; \|hB\| \gg 1, \;\; \|hE\| < 1 \tag{13}$$

where ρ denotes the spectral radius.

For comparison, the classical implicit Euler formula is defined by $M_E = (I - hB)^{-1}$, and both M_D and M_E approximate $\exp(hB)$. The objective of the partitioning is to achieve $M_D \approx M_E$ and at the same time have small blocks in the block diagonal matrix D.

4.1 Matrix error

This section first studies the effect of the splitting and then the difference between M_D and M_E is studied. This type of analysis is mainly suited for the evaluation of a priory partitionings to be chosen among during integration, cf. the approach taken in section 6 Examples in [3].

The analysis does not depend on the mode of decoupled implicit Euler formula, nor does it depend on whether the problem is homogeneous or non-homogeneous.

4.1.1 Splitting

When the linear system (11) is partitioned as $B = D+E$, it is identical to the splitting $Y' = DY + EY$, and the decoupled implicit Euler formula expressed as $Y_n^{[1]} = M_D Y_{n-1}$ then corresponds to an IMEX method first applying the explicit Euler formula to $Y' = EY$ and then the implicit Euler formula to $Y' = DY$.

Notice that the decoupled implicit Euler formula does not rely on a splitting like (8). Besides, neither the higher modes of decoupled implicit Euler nor the decoupled BDF2 formulas [3] can be interpreted as IMEX methods.

The splitting error is,

$$\| \exp(hB) - \exp(hD)\exp(hE)\| = \| \frac{h^2}{2}(ED - DE)\| + \mathcal{O}(h^3)$$

The principal local truncation errors of implicit and explicit Euler formulas are identical except for the sign. Therefore the splitting error is expected to account for a significant part of the difference between M_D and M_E. However, the leading term of the splitting error given above may not be a very good approximation of the total splitting error.

4.1.2 Error analysis

Define the matrix difference,

$$\Delta = M_E - M_D$$

The difference between the solutions obtained by the classical and decoupled Euler formulas is called the decoupling error, and it can be expressed as follows,

$$Y_n - Y_n^{[1]} = (M_E - M_D)Y_{n-1} = \Delta M_E^{-1} Y_n$$

and the relative error estimated by,

$$\|Y_n - Y_n^{[1]}\| / \|Y_n\| \leq \|\Delta M_E^{-1}\| \tag{14}$$

Alternatively,

$$Y_n - Y_n^{[1]} = \Delta M_E^{-1} Y_n \Leftrightarrow M_E^{-1}(Y_n - Y_n^{[1]}) = M_E^{-1}\Delta M_E^{-1} Y_n \tag{15}$$

and the estimate,

$$\|Y_n - Y_n^{[1]}\|_M / \|Y_n\|_M \leq \|M_E^{-1}\Delta\|_\infty \tag{16}$$

with the norm definition $\|v\|_M = \|M_E^{-1}v\|_\infty$.

It is not obvious that the estimate (16) should be sharper than (14). However, this is usually observed during the integration of the smooth solution of a stiff problem where relations (13) apply. Since Δ and M_E^{-1} generally do not commute, the estimates cannot be expected to be equal either.

The matrix $(I - hD)^{-1}$ can be approximated as follows,

$$\begin{aligned}(I - h(B - E))^{-1} &= (M_E^{-1} + hE)^{-1} = M_E(I + hEM_E)^{-1} \\ &= M_E(I - hEM_E) + \mathcal{D}_1\end{aligned} \tag{17}$$

where $\|\mathcal{D}_1\| = \mathcal{O}(h^2\|(EM_E)^2\|)$. We then have

$$M_D = (M_E(I - hEM_E) + \mathcal{D}_1)(I + hE) = M_E(I + hE(I - M_E)) + \mathcal{D}_2 \tag{18}$$

where $\|\mathcal{D}_2\| = \mathcal{O}(h^2\|(EM_E)^2\|)$.

The matrix difference Δ can be expressed as

$$\Delta = M_E - M_D = h^2(I - hD)^{-1}EBM_E = hM_EE(M_E - I)) + \mathcal{D}_2$$

using (18). We now have

$$M_E^{-1}\Delta \approx hE(M_E - I) \quad \text{and} \quad \Delta M_E^{-1} = h^2(I - hD)^{-1}EB \tag{19}$$

The approximate expressions for $M_E^{-1}\Delta$ and ΔM_E^{-1} can be bounded as follows,

$$\|hE(M_E - I)\| \leq \|hE\|(\|M_E\| + 1) \quad \text{and} \quad \|h^2M_EEB\| \leq \|M_E\|\|hE\|\|hB\|$$

The relations (13) imply $\|hE\|(\|M_E\|+1) \ll \|M_E\|\|hE\|\|hB\|$, and therefore it is also to be expected that $\|M_E^{-1}\Delta\| \ll \|\Delta M_E^{-1}\|$.

During the transient solution h is adjusted such that $\|hB\| \ll 1$. With $\|\Delta\| = \mathcal{O}(h^2)$ and

$$\|M_E^{-1}\Delta - \Delta M_E^{-1}\| = \|h(\Delta B - B\Delta)\| = \mathcal{O}(h^3)$$

it follow that $\|M_E^{-1}\Delta\| = \mathcal{O}(h^2)$, $\|\Delta M_E^{-1}\| = \mathcal{O}(h^2)$ and $M_E^{-1}\Delta \to \Delta M_E^{-1}$ for $h \to 0$. These results agree completely with the results in [3].

During an iterative process for obtaining an effective partitioning, $hE(M_E - I) \approx M_E^{-1}\Delta$ is useful for evaluating the accuracy of the partitioning. The matrix M_E may be expensive to compute, but it is only needed once since it is independent of the partitioning.

Example 2

With the partitioned problem from the first part of example 1, D=diag(B_{11}, B_{22}), $E = B - D$ and $h = 0.1$, we get the following value of the leading term of the splitting error $(h^2/2)\|ED - DE\| = 0.585$, while the splitting error evaluates to $\|\exp(hB) - \exp(hD)\exp(hE)\| = 0.2687$.

For the same values, the scaled matrix difference can be computed to

$$\|M_E^{-1}\Delta\|_\infty = 0.55 \quad \approx \|hE(M_E - I)\|_\infty = 0.5217$$

The following table illustrates the relations between $M_E^{-1}\Delta$, ΔM_E^{-1} and step size h,

h	0.01	0.1	1
$\|M_E^{-1}\Delta\|_\infty$	0.01	0.55	10
$\|\Delta M_E^{-1}\|_\infty$	0.01078	0.9167	36.67

In this example it is clear that $\|M_E^{-1}\Delta\|_\infty$ always gives the sharper bound.

4.2 Solution error

When the decoupled implicit Euler formula is used adaptively, it is desirable and also natural to use an estimate of the decoupling error to control the partitioning. However, such an estimate will typically depend on both the mode of the decoupled Euler formula and on whether the problem is homogeneous or non-homogeneous.

The analysis in section 4.2 is basically for homogeneous problems and mode 1 of the decoupled Euler formula. When the analysis applies wider, it is noted, and some of the analysis is extended to non-homogeneous problems and/or higher modes.

4.2.1 Direct computation

The matrix norms considered so far lead to bounds like (16) which are often fairly pessimistic. Alternatively a vector error like (15) may be considered. The left hand side can be estimated using (19) as follows,

$$M_E^{-1}(Y_n - Y_n^{[1]}) \approx hE(M_E - I)Y_{n-1} = hE(Y_n - Y_{n-1}) \qquad (20)$$

Now follows an estimate which is valid for a linearization of (1) along the solution,

$$Y' = B_n(Y - \tilde{Y_n}) + F_n \qquad (21)$$

where Y is assumed to include t as the last element and

$$B_n = \begin{pmatrix} \frac{\partial F}{\partial Y}|_{(t_n,\tilde{Y}_n)} & \frac{\partial F}{\partial t}|_{(t_n,\tilde{Y}_n)} \\ 0^T & 0 \end{pmatrix}, \quad F_n = \begin{pmatrix} F(\tilde{Y}_n) \\ 1 \end{pmatrix}$$

The classical implicit Euler formula applied to (21) gives

$$Y_n = Y_{n-1} + h[B_n(Y_n - \tilde{Y}_n) + F_n]$$

and the decoupled implicit Euler formula, for $B_n = D + E$, gives,

$$Y_n^{[1]} = Y_{n-1} + h[D(Y_n^{[1]} - \tilde{Y}_n) + F_n] \tag{22}$$

The identically zero term $E(\tilde{Y}_n - \tilde{Y}_n)$ was omitted. Subtraction gives,

$$Y_n - Y_n^{[1]} = h[B_n(Y_n - Y_n^{[1]} + Y_n^{[1]} - \tilde{Y}_n)) - D(Y_n^{[1]} - \tilde{Y}_n)]$$

which solves to,

$$Y_n - Y_n^{[1]} = (I - hB_n)^{-1}hE(Y_n^{[1]} - \tilde{Y}_n) \tag{23}$$

The difference $Y_n^{[1]} - \tilde{Y}_n$ is readily computed from (22),

$$Y_n^{[1]} - \tilde{Y}_n = (I - hD)^{-1}[Y_{n-1} + hF_n - \tilde{Y}_n] \tag{24}$$

In the non-autonomous case, it is easy to choose $\tilde{Y}_n$ to obtain a zero as the last element of $Y_n^{[1]} - \tilde{Y}_n$ and $Y_{n-1} + hF_n - \tilde{Y}_n$. With the last row of B_n being equal to zero, the analysis, and notably (23) and (24), can just as well be performed for the linearized system with $B_n = \partial F/\partial Y$, neglecting the equation for t, $t' = 1$.

Notice that $\|Y_{n-1} + hF_n - Y_n\| = \mathcal{O}(h^2)$ and $\|\tilde{Y}_n - Y_n\| = \mathcal{O}(h)$ or $\mathcal{O}(h^2)$ depending on whether mode 1 or mode 2 is used (cf. [3]). This implies that $\|Y_n - Y_n^{[1]}\| = \mathcal{O}(h^2)$ or $\mathcal{O}(h^3)$, respectively. This agrees fully with the results in sections 4.3.1 and 4.3.2 in [3].

The approach, with obvious modifications, also applies to the decoupled BDF2.

4.2.2 Iteration

The decoupled implicit Euler formula (12) can be reformulated into an iteration converging towards the classical Euler formula,

$$Y_n^{[m+1]} = Y_{n-1} + h(DY_n^{[m+1]} + EY_n^{[m]}), \quad Y_n^{[0]} = Y_{n-1} \tag{25}$$

The iteration matrix is then $G = (I - hD)^{-1}hE$. A small norm of G results in a fast convergence of the iteration process and a small error of $Y_n^{[1]}$ (12),

$$\|Y_n^{[1]} - Y_n\| \leq \frac{\|G\|}{1 - \|G\|}\|Y_n^{[1]} - Y_n^{[0]}\| \quad \text{for} \quad \|G\| < 1 \tag{26}$$

This error bound also applies to the non-homogeneous case (11).

The iteration matrix G can be approximated using (18), $G \approx hM_E(I - hEM_E)E$. However, the complexity of this matrix computation is dominated by a $6S^3$ term, and it is therefore three times as expensive to compute as $hE(M_E - I)$. An alternative is to use the crude approximation $G \approx hM_EE$.

The convergence rate k_m can be computed as,

$$k_m = \|Y_n^{[m+1]} - Y_n^{[m]}\|/\|Y_n^{[m]} - Y_n^{[m-1]}\| \tag{27}$$

where $k_m \leq \|G\|$. With the usual convergence criterion of the power method for computing the dominant eigenvalue, we have $k_m \to \rho(G)$ for $m \to \infty$. Therefore k_m may be used in a – somewhat uncertain – approximation of the iteration error, e.g.

$$\begin{aligned} \|Y_n^{[1]} - Y_n\| &\approx \frac{k_1}{1 - k_1}\|Y_n^{[1]} - Y_n^{[0]}\| \quad \text{for} \quad k_1 < 1 \\ &\approx \|Y_n^{[2]} - Y_n^{[1]}\| \quad \text{for} \quad k_1 \ll 1 \end{aligned} \tag{28}$$

The value of k_1 is used for evaluating a partitioning in the implementation described in section 5.2 in [3].

With the decoupled implicit Euler formula (12), the computation of k_1 requires an extra iteration, i.e. an extra solution using the factorized matrix $I - hD$.

Notice that (28) can be used independently of the mode of the decoupled Euler formula and with the decoupled BDF2 formula as well (using the appropriate BDF2 formula for (25)).

4.2.3 Residual

The residual

$$r_n^{[1]} = Y_n^{[1]} - Y_{n-1} - hBY_n^{[1]} = (I - hB)Y_n^{[1]} - Y_{n-1} = M_E^{-1}(M_D - M_E)Y_{n-1}$$

is easily computed, and it can be used for approximating the error

$$Y_n^{[1]} - Y_n = M_Er_n^{[1]} \approx (I - hD)^{-1}r_n^{[1]} \tag{29}$$

This relation also applies to the non-homogeneous problem (11) with the appropriate definition of $r_n^{[1]}$, since (29) simply defines a Newton step.

The approximation is somewhat crude since $(I-hD)^{-1} \approx M_E(I-hEM_E)$ according to the better approximation (18).

Notice that (29) applies independently of the applied mode of the decoupled Euler formula. It also applies for the decoupled BDF2 formula with the obvious substitution for M_E and appropriate definition of $r_n^{[1]}$.

The relative error defined in (16) can be computed as,

$$\|Y_n - Y_n^{[1]}\|_M / \|Y_n\|_M = \|r_n^{[1]}\|_\infty / \|Y_{n-1}\|_\infty \tag{30}$$

Error estimates based on the residual are best suited for being used for monitoring the error along the solution.

The estimate (30) only costs the norm computations on top of the residual computation.

Example 3

With $Y_0 = Y(1)$ given in example 1, we find using the partitioning, the value of $\|M_E^{-1}\Delta\|_\infty$ and other values from example 2 that (16) gives $\|Y_1 - Y_1^{[1]}\|_M/\|Y_1\|_M \leq 0.55$ while (20) results in $\|Y_1 - Y_1^{[1]}\|_M/\|Y_1\|_M \approx \|hE(Y_1 - Y_0)\|/\|Y_0\| = 0.0091$. However, direct computation gives $\|Y_1 - Y_1^{[1]}\|_M/\|Y_1\|_M = 0.0075$.

For G defined in section 4.2.2 we find $\|G\| = 0.8333$, $\rho(G) = 0.2041$ and from (27), $k_1 = \|Y_1^{[2]} - Y_1^{[1]}\|/\|Y_1^{[1]} - Y_0\| = 0.055$. We can bound the decoupling error using (26), $\|Y_1^{[1]} - Y_1\| \leq 0.29$. The actual error is $\|Y_1^{[1]} - Y_1\| = 5.76_{10} - 3$, so the bound is rather conservative. Using the estimate (28), we get $\|Y_1^{[1]} - Y_1\| \approx 3.33_{10} - 3$ which is obviously not a bound but an estimate of the correct order of magnitude.

Using the residual, we find for (30) $\|r_1^{[1]}\|_\infty/\|Y_0\|_\infty = 0.0075$ which is identical to the value of $\|Y_1 - Y_1^{[1]}\|_M/\|Y_1\|_M$ computed above. The approximation (29) leads to

$$\|Y_1^{[1]} - Y_1\| = 5.7633_{10} - 3 \approx \|(I - hD)^{-1} r_1^{[1]}\| = 3.1440_{10} - 3$$

5 Partitioning

In this section the matrix B is assumed to be the Jacobian of F as defined in (11). It will be assumed that this matrix is available both for determining a partitioning and for some of the formulas for evaluating the accuracy of the decoupled integration formula.

The objective is to devise an algorithm which reorders B such that the same reordering applied to the equations and variables of the original system (1) results in a partitioned system (5) which can be solved efficiently and accurately using the decoupled implicit Euler method either in its basic form (6), in one of the more accurate forms or using the decoupled BDF2 [3].

5.1 Regular splitting

Consider a partitioning of B, $B = D + E$ and the corresponding splitting $(I - hD) - hE$ of $I - hB$. This splitting is called regular [6] if $(I - hD)^{-1} \geq 0$ and $hE \geq 0$. The splitting is convergent, i.e. $\rho((I - hD)^{-1}hE) < 1$, if $(I - hB)^{-1} \geq 0$ and the splitting is regular.

Consider two different splittings, $I - hD_1 - hE_1 = I - hD_2 - hE_2$ and assume that they both are regular and convergent. Then the following comparison theorem follows [6],

$$0 \leq E_1 \leq E_2 \Rightarrow \rho((I - hD_1)^{-1}hE_1) < \rho((I - hD_2)^{-1}hE_2) < 1$$

for $E_1 \neq 0$ and $E_1 - E_2 \neq 0$.

Referring to the discussion in connection with the approximation (28) in section 4.2.2, $\rho((I - hD)^{-1}hE)$ is a good measure of the accuracy of the splitting. The highest computational efficiency is obtained with E being as close as possible to the off-diagonal part of B. However, this may lead to a value of $\rho((I - hD)^{-1}hE)$ which is unacceptably large. The objective of the partitioning algorithm is to find the best compromise.

The comparison theorem is the theoretical basis for the partitioning strategy. Assume that $I - hB$ is an M-matrix. Consider a sequence of partitionings of B, $B = D_1 + E_1 = D_2 + E_2 = ...$ where E_1 consists of the numerically smallest (non-zero) off-diagonal element of B, E_2 consists of the two numerically smallest (non-zero) off-diagonal elements of B, etc. Hence $E_i \leq E_j$ for $i < j$.

The splittings $(I - hD_i) - hE_i$, $i = 1, 2, ...$ are all regular and convergent because $I - hB$ is assumed to be an M-matrix [6], and the comparison theorem therefore applies.

The partitioning of the Jacobian of a practical problem does not necessarily lead to a regular splitting $(I - hD) - hE$ which is convergent, at least not for step sizes h of practical interest. The paper [7] presents weaker splittings for which comparison theorems also exist. However, the partitioning algorithm may be applied to problems where $E_i < E_j$ results in $\rho((I - hD_i)^{-1}hE_i) > \rho((I - hD_j)^{-1}hE_j)$, and the partitioning algorithm must take this into account.

Example 4

Continuing example 3 we find that $I - hB$ is an M-matrix for $h \geq 0$. With D_1 =diag(B_{11}, B_{22}) as in example 4 and $D_2 = \begin{pmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{pmatrix}$, we have regular and convergent splittings corresponding to the partitionings $B = D_1 + E_1 = D_2 + E_2$ where $\max|E_1| = 10$ and $\max|E_2| = 1$. For $h = 0.1$ we can compute

$$\rho((I - hD_1)^{-1}hE_1) = 0.2041 > \rho((I - hD_2)^{-1}hE_2) = 0.0417$$

5.2 Symmetric reordering

The partitioning of a system of differential equations is based on a matrix reordering algorithm which uses a symmetric row–column reordering specified by the permutation matrix P to arrive at a lower block triangular matrix,

$$\tilde{B} = PBP^T = \begin{pmatrix} B_{11} & 0 & ... & 0 \\ B_{21} & B_{22} & ... & 0 \\ ... & ... & ... & ... \\ B_{q1} & B_{q2} & ... & B_{qq} \end{pmatrix}$$

In [8] an efficient algorithm for performing this reordering is described. The complexity of the algorithm is $8(S+NZ)+64S$ where NZ is the number of non-zero elements of B and S is the dimension.

If B is symmetric, the result of the reordering is a block diagonal matrix ($B_{ij} = 0$ for $i \neq j$). Obviously the reordering may fail to produce more than one diagonal block which is then the original matrix, possibly reordered to some extent.

5.3 Elements of a partitioning algorithm

The objective of the partitioning algorithm is to obtain a reordering and partitioning of B into $\tilde{B} = D + E$ where $\max|E| < \delta$ and D is a lower block triangular matrix. The maximum absolute value of a matrix is defined as $\max|E| = \max_{i,j}|e_{ij}|$. The reordering is performed as follows.

> **δ-partitioning.**
> Delete all off-diagonal elements of B numerically smaller than δ to obtain B_δ. Use a symmetric row–column reordering [8] on B_δ to obtain the lower block triangular matrix $\tilde{B}_\delta$.
>
> Apply the same reordering to B resulting in $\tilde{B}$ which can then readily be partitioned into a lower block triangular matrix D with

the same envelope as $\tilde{B}_\delta$ and the upper block triangular (without diagonal blocks) part E where $\max|E| < \delta$.

The partitioning where D is a lower block triangular form is optimal for execution on a sequential processor. For parallel processing D should be block diagonal. This is easily obtained by performing the reordering to block form of the matrix $B_\delta + B_\delta^T$.

In order to obtain diagonal blocks of the same size as with the lower block triangular form, is will usually be necessary to choose a larger value of δ resulting in a less accurate splitting.

When $I - hB$ is an M-matrix, a splitting with $\rho((I - hD)^{-1}hE) \approx \rho_0$ can be obtained by applying a secant type iteration to $g(\delta) = \rho((I - hD(\delta))^{-1}hE(\delta))$ where $B = D(\delta) + E(\delta)$ denote a δ-partitioning of B. The iteration is complicated by the fact that $g(\delta)$ is not continuous. However, according to the comparison theorem, $\delta_i < \delta_j \Rightarrow g(\delta_i) \leq g(\delta_j)$.

The purpose of the partitioning is to reduce the computational complexity from that of a traditional approach given in (4) to the complexity of the decoupled Euler method (7). The optimal partitioning results in a D-matrix with as many diagonal blocks q of uniform dimension as possible. Obviously $q \leq S$.

As long as accuracy is only sacrificed insignificantly by the use of the decoupled Euler method, the reduction in computational complexity is a complete gain. If the partitioning leads to a reduced accuracy which must be compensated by an extra iteration in (6) or a shorter step size, the situation is more complicated.

It is possible to model the computational cost of applying the decoupled Euler method to a given problem fairly accurately. This can be used in an outer loop of the partitioning algorithm where ρ_0 is adjusted to either reduce or increase the number and size of the diagonal blocks of $D(\delta)$.

The partitioning algorithm has some drawbacks. First, it assumes that $I - hB$ is an M-matrix. This will often *not* be the case in practical problems although $I - hB$ may be "close" to being an M-matrix. Besides, it may be expensive to establish whether $I - hB$ is an M-matrix or not. When $I - hB$ is not an M-matrix, the comparison theorem does not hold, and the partitioning algorithm becomes more complicated.

Second, the measure $g(\delta) = \rho((I - hD(\delta))^{-1}hE(\delta))$ is expensive to compute. However, a spectral radius approximation based on the power method may suffice.

Third, the relation between $g(\delta)$ and the accuracy of the solution of the decoupled Euler is only simple when $g(\delta) \approx k_1$, cf. (27), such that the approximation (28) applies.

The system (1) is in general nonlinear, and the Jacobian B is computed at some point Y_n along the solution (possibly the initial value Y_0). The partitioning algorithm to be proposed is iterative so therefore the evaluation of the partitioning has to be performed several times. It should therefore be as computationally cheap as possible.

The norm of the approximation $hE(M_E - I)$ of the matrix $M_E^{-1}\Delta$ (19) has been used successfully. The expensive part of this computation is M_E which is independent of the partitioning. Alternatively – and even cheaper – is the use of the vector error estimate obtained from (20),

$$\|Y_n - Y_n^{[1]}\|_M / \|Y_n\|_M \approx \|hE(Y_n - Y_{n-1})\| / \|Y_{n-1}\| \tag{31}$$

The choice of the matrix norm as acceptance criterion is the more conservative which will be valid along the solution as long as the Jacobian has not changed too much. The vector error norm estimate gives a fairly accurate local value corresponding to the current solution vector, cf. example 3. However, it only applies to the decoupled Euler formula, mode 1.

The approximation in (31) can be used to obtain an initial value of the threshold δ for the partitioning algorithm,

$$\delta_0 = \varepsilon_{tol} \|Y_{n-1}\| / \|h(Y_n - Y_{n-1})\|$$

where ε_{tol} is the local truncation error tolerance.

5.4 An example of a partitioning algorithm

Some notation is introduced for the partitioning algorithm. The control structure used is close to that in the programming language C, but otherwise a matrix-vector notation close to that of the rest of the paper is used. Scalars are denoted by Greek letters except n. A partitioning is described by $\mathcal{R}$ and $\mathcal{E}$ where $\mathcal{R}$ is a permutation matrix and the matrix-valued function $\mathcal{E}$ deletes the D-part of a matrix thus resulting in the E-matrix.

A partitioning is characterized by the total area of the diagonal blocks, $\rho = \sum_{r=1}^{q} s_r^2$ and the partitioning error of the linearized problem, Φ. All of this information is kept in the data structure $\mathrm{P} = \{[\mathcal{R}, \mathcal{E}], \Phi, \rho\}$. The value of a field is denoted by the usual dot-notation, say $\mathrm{P}.\Phi$.

P_n is at most updated every 10 steps (n is the step number), and it is assumed that $\mathrm{P}_n = \mathrm{P}_{n-1}$ unless P_n is updated by the partitioning algorithm.

The following partitioning algorithm is used in the program generating the results of the example problem in section 6. It is invoked after each 10 iteration steps depending on ϕ_n, the estimate of the partitioning error derived from an extra iteration of the decoupled implicit Euler formula ((2.9) in [3]) leading to $Y_n^{[2]}$ cf. (28).

```
1   $\phi_n = \|Y_n^{[2]} - Y_n^{[1]}\|$;
2   **if** $(n$ **mod** $10 == 0 \wedge (\phi_n > 5\varepsilon_{tol} \vee (\phi_n < \varepsilon_{tol}/5 \wedge \mathrm{P}_n.\rho > 0)))$ {
3       $\Delta\tilde{Y}_n^{[1]} = (I - hD_n)^{-1}(Y_{n-1} - hF(\tilde{Y}_n) - \tilde{Y}_n)$;
4       **if** $(\phi_n > 5\varepsilon_{tol})$  $\mathrm{P}_{n+1} = \{[\mathcal{I}, \mathcal{O}], 0, S^2\}$;
5          **else**  { $\mathrm{P}_{n+1} = \mathrm{P}_n$; $\mathrm{P}_{n+1}.\Phi = \phi_n$; }
6       $\sigma = 1$;  $\Phi_0 = \phi_n$;   $[\mathcal{R}_0, \mathcal{E}_0] = \mathrm{P}_n.[\mathcal{R}, \mathcal{E}]$;
7       $\delta_1 = \max|\mathcal{E}_0(\mathcal{R}_0 B_n \mathcal{R}_0)|\sqrt{\varepsilon_{tol}/\phi_n}$;
8       **for** $(i = 1;\ i \leq 3;\ i{+}{+})$ {
9          **create** $B_{\delta_i}$;
10         $[\mathcal{R}_i, \mathcal{E}_i, \rho_i] =$ **reorder** $(B_{\delta_i})$;
11         $\Phi_i = \|(I - hD_n)^{-1} h\mathcal{R}_i\mathcal{E}_i(\mathcal{R}_i D_n \mathcal{R}_i)\mathcal{R}_i\Delta\tilde{Y}_n^{[1]}\|$ ;
12         $\mathrm{P}^{[i]} = \{\ [\mathcal{R}_i, \mathcal{E}_i], \Phi_i, \rho_i\}$;
13         **if** $((\rho_i == \mathrm{P}_{n+1}.\rho \wedge \Phi_i < \mathrm{P}_{n+1}.\Phi) \vee (\rho_i < \mathrm{P}_{n+1}.\rho \wedge \Phi_i < 5\varepsilon_{tol}))$
14            $\mathrm{P}_{n+1} = \mathrm{P}^{[i]}$;
15         **if** $(\mathrm{P}_{n+1}.\Phi < 5\varepsilon_{tol} \wedge (\mathrm{P}_{n+1}.\Phi > \varepsilon_{tol}/5 \vee \mathrm{P}_{n+1}.\rho == 0))$   **exit** i-loop;
16         **if** $(\Phi_{i-1} == \Phi_i)$   $\sigma = \sigma\varepsilon_{tol}/\Phi_i$;   **else**   $\sigma = \sqrt{\varepsilon_{tol}/\Phi_i}$;
17         **if** $(i == 2 \wedge ((\Phi_{i-1} > \varepsilon_{tol} \wedge \Phi_i < \varepsilon_{tol}) \vee (\Phi_{i-1} < \varepsilon_{tol} \wedge \Phi_i > \varepsilon_{tol})))$
18            $\delta_{i+1} = \sqrt{\delta_i\delta_{i-1}}$;   **else**   $\delta_{i+1} = \sigma\ \max|\mathcal{E}_i(\mathcal{R}_i B_n \mathcal{R}_i)|$;
19      }
20  }
```

The partitioning algorithm has three main components, the computation of δ, the δ-partitioning and the evaluation of the resulting partitioning based on (23) and (24).

Because of the problems mentioned in the end of section 5.3, the partitioning algorithm is more a search algorithm than an iteration algorithm. The partitioning is initialized to assure that the accuracy is adequate, $\mathrm{P}_{n+1}.\Phi < 5\varepsilon_{tol}$, and then at most three iterations towards a better partitioning are performed.

While the accuracy is measured by ϕ and Φ, the efficiency of the partitioning is measured by ρ, the sum of the areas of the diagonal blocks. A partitioning resulting in S scalar subsystems is characterized by $\rho = 0$.

The value ε_{tol} is the set-value for the control of the local truncation error, and the aim of the partitioning algorithm is to keep the partitioning error ϕ_n close to ε_{tol} to maintain accuracy (small ϕ_n) and efficiency (small ρ). The partitioning algorithm is now commented in detail.

Line 2: The partitioning is evaluated every 10 steps and possibly changed. If the partitioning is changed too frequently, it results in oscillations in the solution.

Line 3: Approximate computation of the difference in (24).

Line 4 and 5: If the error of the current partitioning is too big, the initial partitioning of the search is taken to be no partitioning characterized by an identity permutation matrix and a zero-function $\mathcal{O}$ to generate the E-matrix. Otherwise the current partitioning is chosen in line 5.

Line 7: δ for the δ-partitioning is computed aiming at obtaining $\phi_{n+1} = \varepsilon_{tol}$. The definition of Φ_i in line 11 indicates a linear relationship between $\mathcal{E}_i(\mathcal{R}_i B_n \mathcal{R}_i)$ and Φ_i, but the square root is used in the formulas in line 7 and 16 in order to reduce the excursions of δ and avoid oscillations in the partitioning algorithm.

Line 9 and 10 essentially specify the δ-partitioning algorithm described in section 5.3.

Line 11: Decoupling error using an approximation of (23).

Line 13: The new partitioning $\mathrm{P}^{[i]}$ is accepted if either it is more accurate ($\Phi_i < \mathrm{P}_{n+1}.\Phi$) or more efficient ($\rho_i < \mathrm{P}_{n+1}.\rho$) than the current partitioning.

Line 16: The change in δ is generally determined by $\sigma = \sqrt{\varepsilon_{tol}/\Phi_i}$. If the partitioning algorithm is "stuck", σ is changed to the larger value $\sigma = \varepsilon_{tol}/\Phi_i$.

Line 17 and 18: If Φ_i oscillates out of the bound $[0.2\varepsilon_{tol}, 5\varepsilon_{tol}]$, δ is adjusted to the geometric mean of the previous values.

Remark to line 3 and 11: These lines implement approximations of formula (24) and (23), respectively. In line 3, $\Delta\tilde{Y}_n^{[1]}$ is computed corresponding to D_n. To comply with section 4.2.2 $\Delta\tilde{Y}_i^{[1]}$ should be computed for each new partitioning using $(I - hD_i)^{-1}$ where $D_i = B_n - \mathcal{R}_i \mathcal{E}_i(\mathcal{R}_i B_n \mathcal{R}_i)\mathcal{R}_i$. However, the factorization $(I - hD_n)^{-1}$ is available "for free" since it was used in the current integration step. Likewise this factorization is used in line 11 in stead of $(I - hB_n)^{-1}$. The difference between these matrices appears from (18).

6 Example: Chemical reaction kinetics

The example problem is the same as the problem in [3], a system of 32 ODEs from chemical reaction kinetics used in an air pollution model, see [9] and [10] for more details.

The system has the following structure,

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

The nonlinearities are mainly products, i.e. P_i and L_i are typically sums of terms of the form $c_{ilm}(t)Y_l Y_m$ and $d_{il}(t)Y_l$. The time dependency in $c_{ilm}(t)$ and $d_{il}(t)$ is due to the influence from the sun.

In most equations (all but 4), $L_i(t, Y)$ does *not* depend on Y_i, and by definition, $P_i(t, Y)$ never depends on Y_i. This means that a decomposition of the problem down to scalar equations in most cases results in *linear* discretized equations, resulting in a considerable saving on top of the decomposition itself.

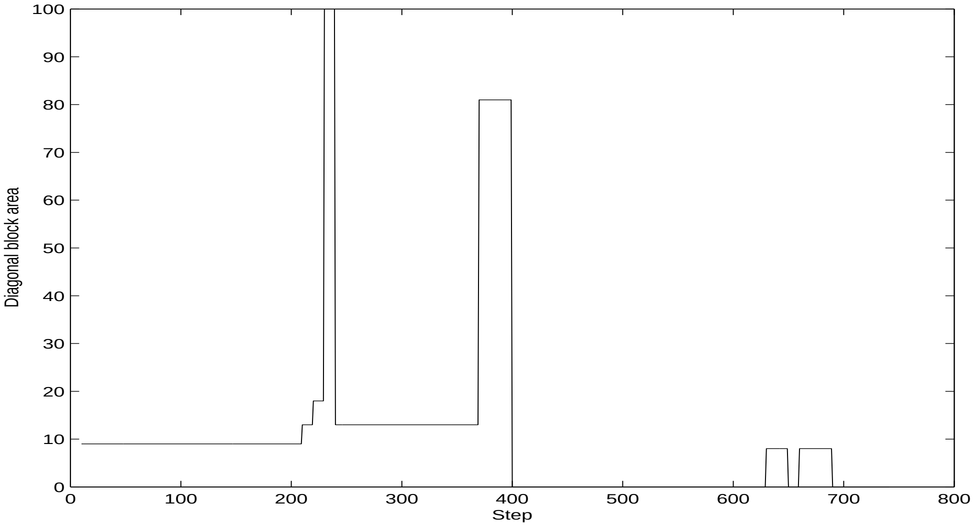

Figure 1: The variation in diagonal block area along the solution.

The example problem is solved using an implementation of the decoupled implicit Euler formula similar to that described in section 5.2 in [3], notably regarding the error estimation and step size control part. The main difference is that the partitioning algorithm from section 5.4 is used here. Besides, mode 2 is always used, and a second relaxation is only used every 10 steps as explained in section 5.4.

The problem is solved over the time interval $[2.16_{10}4, 1.728_{10}5]$ The step size control aims at keeping the local truncation error at $\varepsilon_{tol} = 10^{-3}$. However, the step size is not permitted to drop below 90, and when this limit is reached, the local truncation error becomes greater than 10^{-3}.

Figure 1 shows the result of the partitioning algorithm. The total number of steps is 741, and 292 steps are taken using a partitioning into scalar equations (block area, $\rho = 0$). The variation in diagonal block area along the solution involves partitionings with the following diagonal blocks, $\{3 \times 3\}$, $\{3 \times 3,\ 2 \times 2\}$, $\{3 \times 3,\ 3 \times 3\}$, $\{10 \times 10\}$, $\{3 \times 3,\ 2 \times 2\}$, $\{9 \times 9\}$, and after an interval with a scalar diagonal, $\{2 \times 2,\ 2 \times 2\}$. The initial 10 steps taken with the full system (block area, $\rho = 1024$) are not shown.

With $\varepsilon_{tol} = 10^{-2}$ the partitioning chooses diagonal blocks $\{2 \times 2\}$ and $\{2 \times 2,\ 2 \times 2\}$. Approximately half of the steps are taken using a partitioning into scalar subsytems.

With $\varepsilon_{tol} = 10^{-4}$ and a lower bound of 9 on the stepsize, the partitioning is among the same group of diagonal blocks as for $\varepsilon_{tol} = 10^{-2}$, only now more than 80% of the steps are taken using a partitioning into scalar subsytems.

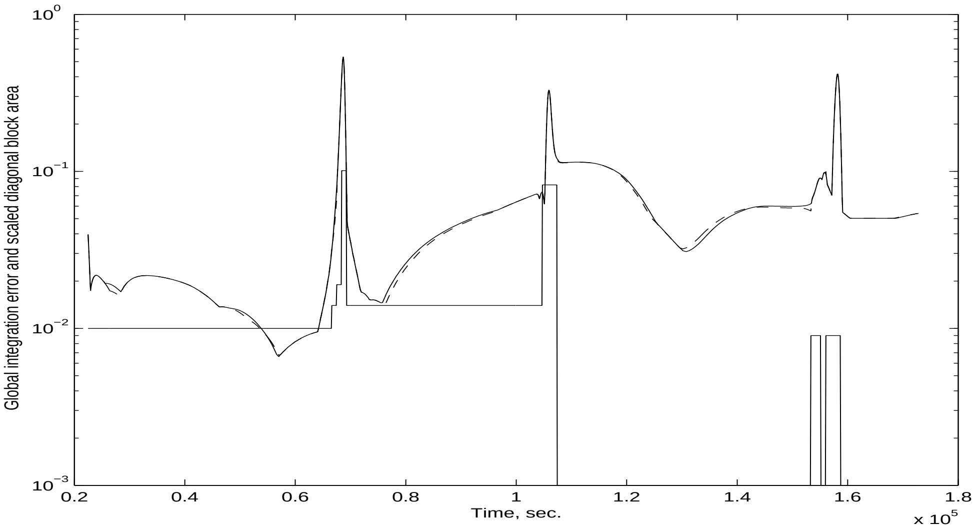

Figure 2: Global integration error, Euler: solid line, decoupled Euler: dashed line. Scaled diagonal block area: lower solid line.

Figure 2 compares the global errors obtained by the classical implicit Euler formula and the decoupled Euler formula. The classical Euler formula is applied with the same step size selection as the decoupled Euler formula. The discrepancy between the errors is seen to be insignificant.

The global error is obtained by comparing with a reference solution computed using a variable step size variable order (maximum order = 6) implementation of the backward differentiation formulas [11] with a bound on the relative local error estimate of $0.01\varepsilon_{tol}$. 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 and global error 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) next day, and 7 p.m. (t=155,000).

The figure also shows the partitioning for reference. The diagonal block area is scaled as follows: $(\rho + 1)\times 10^{-3}$. It is obvious that the the large spikes in the error necessitate more conservative partitionings.

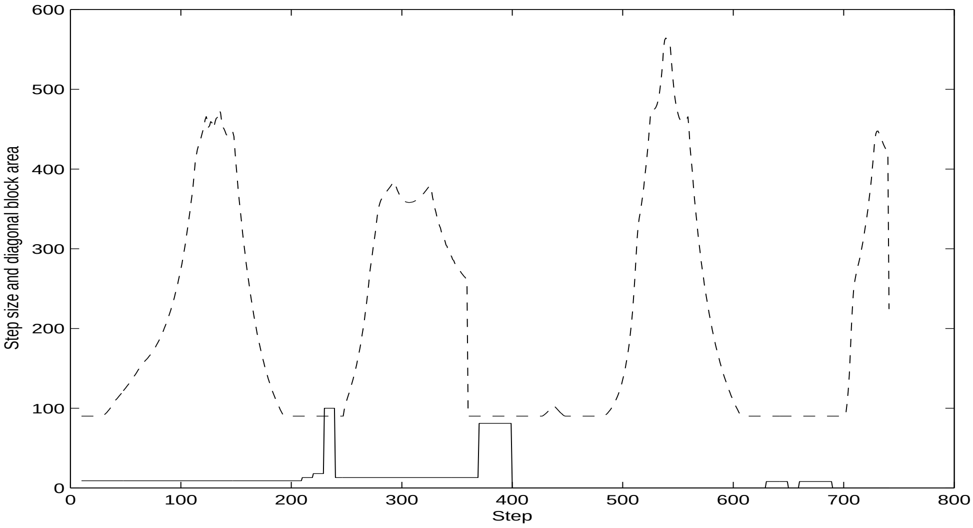

Figure 3: Step size variation: dashed line. Diagonal block area: solid line.

When the stepsize is not bounded from below to 90 but controlled exclusively by the local truncation error estimate, the spikes in the error are less pronounced (maximum 0.12) and the partitioning is into scalar equations for all of the integration after the first 10 steps. Figure 3 shows the stepsize variation and the corresponding partitioning.

Figure 4 shows some of the key data used in controlling the partitioning algorithm. The relaxation error ϕ_n is computed for every step in the experimental algorithm to provide data for the plot in figure 4. However, the value is only used every 10 steps like the partitioning algorithm in section 5.4 shows, and in a production program ϕ_n would only be computed every 10 steps.

Inside the partitioning algorithm, the linearized error Φ_n is computed at least once, and the graph shows the value corresponding to the partitioning being chosen. The value of Φ_n is not changed until the next repartitioning. You would expect ϕ_n and Φ_n to agree reasonably well immediately after Φ_n has been computed, and this is observed in the figure.

With 741 steps in this example, the partitioning is potentially invoked 74 times. However, because of the test in line 2 of the partitioning algorithm, it is only invoked 15 times. Out of these, a new partitioning is found in the first attempt 7 times, and it takes the maximum of three iterations 8 times. In total, re-partitionings are performed $31 = 7 + 8 \times 3$ times.

The system of ODEs is very stiff with the real part of the eigenvalues of the Jacobian along the solution ranging from 0 to $-8_{10}4$ The partitioning and reordering of the Jacobian B results in $\tilde{B} = D + E$ where $\max |E| \leq$

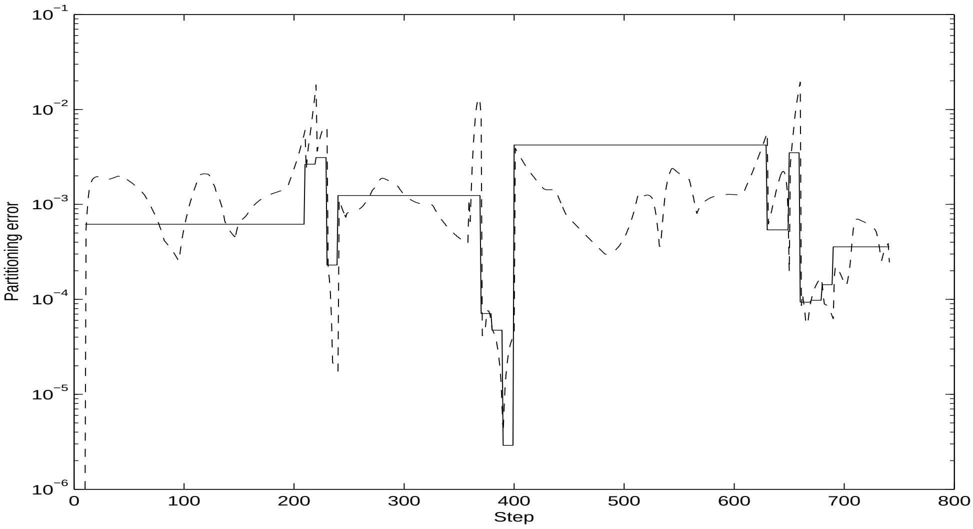

Figure 4: Relaxation error $\phi_n = \|Y_n^{[2]} - Y_n^{[1]}\|$, dashed line. Linearized error Φ_n, solid line.

δ. Consider the linearized decoupled Euler formula (12). With $h\delta < 1$ the stiffness has been confined to the implicit part while the explicit part handles the rest. However, near step $n = 539$ and an upper triangular E corresponding to a decomposition of the problem down to scalar equations, $h \max|E| = 3.13$ peaks and likewise for $n = 730$, we have $h \max|E| = 2.48$

When the example is executed with $\varepsilon_{tol} = 10^{-2}$, $h \max|E|$ reaches a value of 107 for an upper triangular E. This illustrates that the decoupled implicit Euler formula retains its stability even when there is considerable stiffness in the explicitly solved part of the problem.

The purpose of using the decoupled Euler formula in this example is to obtain a more efficient algorithm than the classical implicit Euler formula. An approximate comparison between the algorithms for this example shall now be presented. It is based on the complexity formulas (4) and (7) with the addition of the complexity of the partitioning algorithm in section 5.4.

Each new step is preceded by the computation of the rate constants $c_{ilm}(t)$ and $d_{il}(t)$, in total 69. This computation involves numerous exponentials and some trigonometric functions. The floating point operation count is estimated to $C_{rate} = 2800$.

The floating point operation counts for the function F and the Jacobian $\partial F/\partial Y$ are 480 and 1670, respectively resulting in the coefficients $\eta_F = 0.47$ and $\eta_J = 1.63$. These values turn out to be small compared to a full LU factorization of a 32×32 matrix, $C_{LU} = 21328$.

The complexity of a step with the classical Euler formula according to (4) with $S = 32$ and $n_{it} = 2$ including the computation of the rate coefficients, amounts to approximately 31300 FLOPs.

The cost of a step of the decoupled Euler formula is evaluated according to (7) with $s_1 = 10$ and $s_r = 1$ for $r > 1$, resulting in $C_{DEul} = 1300$.

Before the partitioning algorithm is entered, the error ϕ_n must test greater than a threshold plus some other conditions. The computation of ϕ_n costs an extra relaxation of the decoupled Euler formula every 10 steps.

The total complexity of the partitioning algorithm is divided into initialization and iteration. Initialization is approximately $C_{Pinit} = (\eta_F + \eta_J)S^2 + \frac{1}{2}S^2 + 3S + C_{Dsolv}$ where $C_{Dsolv} = S^2 + s_1^2$ is the FLOP count for the solution based on a factored block triangular matrix $(I - hD_n)^{-1}$ with just one diagonal block of dimension $s_1 = 10$ in this example. The iteration cost is approximately $C_{Pit} = C_{reord} + C_{Dsolv} + 2S^2$, where the operation count for the block triangular reordering algorithm is $C_{reord} = 4104$ for $NZ = 225$, cf. section 5.2.

Furthermore, the partitioning algorithm is entered 15 times in 741 steps and a total of 31 iterations are performed. This results in an average computational cost per step of 1813 FLOPs according to the following formula, where the factor 1.1 accounts for the relaxation used for computing ϕ_n,

$$1.1\, C_{DEul} + \frac{15}{741} C_{Pinit} + \frac{31}{741} C_{Pit}$$

This value is slightly conservative, since a maximum diagonal block with $s_1 = 10$ is only present in 10 steps.

Add to the worst case value the cost of the computation of the rate function and the total cost is then 4613 FLOPs. The saving in this example over the classical Euler formula is better than a factor $31300/4613 = 6.8$.

The complexity analysis involves some inaccuracies. The rate coefficients include numerous transcendental functions, and the complexity value being used is an estimate. The complexity of the entries of F and $\partial F/\partial Y$ vary widely, but an average value is used. Finally, the terms of $\mathcal{O}(S)$ are only partly accounted for.

The saving would be less if the classical Euler implementation had used an efficient sparse solver, but the partitioning algorithm would also benefit. However, the problem is somewhat small to amortize the overhead required by sparse matrix computation.

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] S. Skelboe, *Accuracy of decoupled implicit integration formulas*, SIAM Journal on Scientific Computing, Volume 21, Number 6, pp. 2206-2224

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

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

[6] R.S. Varga, *Matrix iterative analysis*, Prentice Hall, 1962.

[7] Z.I. Woźnick, *Basic comparison theorems for weak and weaker matrix splittings*, The Electronic Journal of Linear Algebra, Vol. 8, pp. 53–59, 2001.

[8] I. S. Duff, J. K. Reid, *An implementation of Tarjan's algorithm for the block triangularization of a matrix*, ACM Trans. Mathematical Software, 4 (1978), pp. 137–147.

[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 modeling*, J. Geophys. Res., 94 (1989), pp. 12925–12956.

[11] 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.