E.D. de Goede, F.W. Wubs

# Explicit-implicit methods for time-dependent partial differential equations

Department of Numerical Mathematics Report NM-R8703 February

# Explicit-Implicit Methods for Time-Dependent Partial Differential Equations

E.D. de Goede, F.W. Wubs
*Centre for Mathematics and Computer Science*
*P.O. Box 4079, 1009 AB Amsterdam, The Netherlands*

For the integration of partial differential equations we distinguish explicit and implicit integrators. Implicit methods allow large integration steps, but require more storage and are more difficult to implement than explicit methods. However, explicit methods are subject to a restriction on the integration step. In this paper, we introduce explicit-implicit methods, which form a combination of explicit and implicit calculations. For these methods, the impact of varying the explicitness, and thus the implicitness, on the stability is examined.

*1980 Mathematics Subject Classification:* 65M05, 65M10, 65M20.
*Key Words and Phrases* : Partial differential equations, explicit-implicit methods, method of lines, tridiagonal equations, incomplete cyclic reduction, stability.
*Note* : These investigations were sponsored in part by the Netherlands Foundation for Technical Research (STW), future Technical Science Branch Division of the Netherlands Organization for the Advancement of Pure Scientific Research (ZWO).
*Note* : This report will be submitted for publication elsewhere.

## 1. Introduction

For the integration of partial differential equations, we distinguish explicit and implicit methods. In general, implicit methods allow large integration steps, but require more storage and are more difficult to implement than explicit methods. However, explicit methods are subject to a restriction on the integration step, because of stability considerations. Implicit methods are in most cases stable for any integration step. This property may be redundant. Here, we concentrate on explicit-implicit methods, which form a combination of explicit and implicit calculations. The objective of such a combination is always to reduce the computational effort to an acceptable level in such a way that the resulting combination still offers attractive stability properties. For the methods presented in this paper, the stability properties vary with the explicitness, and thus the implicitness, of the calculations.

In this paper, we consider one-dimensional problems only. The resulting methods can also be used in alternating direction methods for multi-dimensional cases. We will construct explicit-implicit methods for partial differential equations of the form

$$\frac{\partial \mathbf{u}}{\partial t}(t,\mathbf{x}) = \mathbf{f}(\mathbf{u},\mathbf{u}_x,\mathbf{u}_{xx},\mathbf{x},t), \quad \mathbf{x} \in \Omega \subset \mathrm{I\!R}, \quad t > 0, \tag{1.1}$$

with appropriate boundary conditions. After replacing, on a uniform grid, the spatial derivatives by discrete approximations, we apply an implicit time integrator for the resulting system of ordinary differential equations. We assume that the time integration leads to a linear tridiagonal system of the

Report NM-R8703
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

form

$$A\mathbf{Z} = \mathbf{B}\,, \qquad (1.2)$$

where $A$ is a square tridiagonal matrix, $\mathbf{Z}$ denotes the unknowns at the advanced time level $n+1$ and $\mathbf{B}$ is a column vector. Next, we separate the uniform grid into two sets of grid points. This choice is determined by stability considerations. Let us assume that the unknowns at the advanced time level on the two sets of grid points are $\mathbf{V}$ and $\mathbf{W}$. The system is now reduced by elimination of $\mathbf{W}$ to a system which involves only the unknowns $\mathbf{V}$. This will be called the reduced system. Next, we approximate the solution for the reduced system by an explicit expression. Once $\mathbf{V}$ is solved, $\mathbf{W}$ is solved by back substitution. By this approach the constructed method is essentially explicit.

A reduced system appears naturally when a few steps of a cyclic reduction method[7] are performed on system (1.2). For the cyclic reduction algorithm, Heller[6] proved the following property : If $A$ in (1.2) satisfies certain diagonal dominance conditions, then the ratio of the off-diagonal elements to the diagonal elements decreases quadratically with each cyclic reduction step. This property is the basis of our approach. It was Hockney's observation that in case of constant diagonals the reduction algorithm could be stopped when the ratio of the off-diagonal elements to the diagonal elements fell below machine precision. Then, the tridiagonal system was essentially diagonal and could be solved without damage to the solution. In many cases this process can be stopped before the mentioned ratio falls below machine precision[6]. The constructed solution method is called incomplete cyclic reduction. It is our approach, to approximate the solution of the reduced system by an explicit expression. Using this approach it is possible to stop the reduction process when the ratio of the off-diagonal elements to the diagonal elements is about a factor $1/6$. For example in Table 5.1, it will be shown that our approach requires less cyclic reduction steps in order to obtain accurate results than using incomplete cyclic reduction without any adaptation.

The main purpose of this paper is to construct explicit-implicit methods which have an acceptable stability behaviour. Only for model problems we were able to derive stability conditions. Using our approach, it appeared that the maximum allowed time step increases linearly with the size of the numerical influence domain for hyperbolic equations. For parabolic equations the maximal time step increases quadratically with the size of the influence domain.

Our method can be applied directly to both hyperbolic and parabolic equations. For a given time step, the reduced system for $\mathbf{V}$ can be chosen in such a way that the method is stable. The method can also be used to solve elliptic equations when a time-stepping approach is used (cf. [13],pp. 148-154).

We think that an important application of these techniques are problems which cannot be stored in the central memory of the computer. In such a case one has to evaluate the solution at a new time level block by block. The size of such a block is limited by the size of the central memory. By the given technique one can use the maximal time step which is possible on such a block. Furthermore, with respect to parallel computing one can partition the domain in a number of blocks which can be spread over the available processors.

With respect to vectorization of the solution process, we propose two possibilities which are known in the literature. The first is a modification of the incomplete cyclic reduction method (see [6]) and the second is a modification of a solution method given by Wang[21]. Both have good vectorizing properties (see [8,12,20,21]). In our approach a slight decrease of computation time can be obtained compared with the complete cyclic reduction method and the method of Wang. This decrease depends on the time step (see Remark 4.5).

In Section 2, we show how a system of equations arising from an implicit scheme can be separated in two subsystems which correspond to the values **V** and **W**, respectively. In Section 3, we derive a method for approximating the implicit scheme for **V** by an explicit scheme. In Section 4, the stability condition for this explicit-implicit method is derived. In Section 5, we show by a number of numerical experiments, the impact of varying the explicitness, and thus the implicitness, on the stability. In our numerical experiments we applied the methods to both hyperbolic and parabolic differential equations.

## 2. Construction of the reduced system

Consider the one-dimensional partial differential equation

$$\mathbf{u}_t = \mathbf{f}(\mathbf{u},\mathbf{u}_x,\mathbf{u}_{xx},\mathbf{x},t), \quad \mathbf{x} \in \Omega \subset \mathbb{R}, \quad t > 0, \tag{2.1}$$

with appropriate boundary conditions. Using the method of lines, (2.1) is space discretized on a uniform grid $\Omega_\Delta := \{ j\Delta x \}_j$ . This gives a system of ordinary differential equations [11]

$$\frac{d}{dt}\mathbf{U} = \mathbf{F}(\mathbf{U},t)\,, \quad t > 0\,, \tag{2.2}$$

where $U_j(t)$ approximates $\mathbf{u}(j\Delta x,t)$ and $\mathbf{F}(\mathbf{U},t)$ is a vector function approximating the right-hand side function. Thereafter, a time integrator is applied to (2.2). We confine ourselves to difference formulae, which involve only two adjacent time levels. For the time integration of (2.2) explicit or implicit time integrators can be used. If the solution of (2.2) varies only slowly in time, then usually implicit time integrators are used, which in most cases are stable for any time step. Hence, for the time discretization of (2.2) we consider the $\theta$-method [13]

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t\,\{\theta\,\mathbf{F}(\mathbf{U}^{n+1},t^{n+1}) + (1-\theta)\,\mathbf{F}(\mathbf{U}^n,t^n)\}\,, \ \tfrac{1}{2} \leqslant \theta \leqslant 1\,, \tag{2.3}$$

where we have the second-order Trapezoidal Rule for $\theta = \frac{1}{2}$ and the Backward Euler method for $\theta = 1$ (see [13]).

The equations in system (2.3) may be nonlinear. In order to obtain a linear system of equations, we introduce the so-called splitting function $\mathbf{G}(\mathbf{Z},\tilde{\mathbf{Z}},t)$ [10]. We choose **G** in such a way that it is linear in its second variable, i.e.

$$\mathbf{G}(\mathbf{Z},\tilde{\mathbf{Z}},t) = J(\mathbf{Z},t)\tilde{\mathbf{Z}} + \mathbf{g}(\mathbf{Z},t)\,. \tag{2.4}$$

$J$ and **g** are chosen so that the splitting condition

$$\mathbf{G}(\mathbf{Z},\mathbf{Z},t) = \mathbf{F}(\mathbf{Z},t)$$

is satisfied. Equation (2.3) is now approximately solved by the iteration process

$$\begin{aligned} &\mathbf{Z}^{(0)} = \mathbf{U}^n \\ &\mathbf{Z}^{(q)} = \mathbf{U}^n + \Delta t\,\{\theta\mathbf{G}(\mathbf{Z}^{(q-1)},\mathbf{Z}^{(q)},t^{n+1}) + (1-\theta)\mathbf{G}(\mathbf{U}^n,\mathbf{U}^n,t^n)\} \quad , \ q = 1,\dots,Q \\ &\mathbf{U}^{n+1} = \mathbf{Z}^{(Q)}\,. \end{aligned} \tag{2.5}$$

In this equation, the iterate $\mathbf{Z}^{(q)}$ has to be solved from a linear system of equations. In order to approximate (2.3) accurately by (2.5), $Q$ has to be chosen large. However, to obtain a second-order accurate scheme, convergence is not needed. For a linear problem (**G** is independent of its first variable) the scheme (2.5) is second-order accurate for $Q = 1$, whereas for a nonlinear problem (2.5) is second-order accurate for $Q \geqslant 2$. For the latter, a system of equations has to be solved at least twice at each time step. In Section 5.4, we will introduce a variant of (2.5) for which only one system of equations has to be solved at each time step.

In order to apply scheme (2.5), we have to solve for all $q$, a linear system of equations of the form

$$(I - \theta \Delta t\, J(\mathbf{Z}^{(q-1)}))\mathbf{Z}^{(q)} = \mathbf{B}\,, \tag{2.6}$$

where

$$\mathbf{B} = \mathbf{Z}^{(0)} + \Delta t\, \{\theta \mathbf{g}(\mathbf{Z}^{(q-1)}, t^{n+1}) + (1-\theta)\, \mathbf{G}(\mathbf{U}^n, \mathbf{U}^n, t^n)\}\,.$$

In the following, we will simply write $J$ instead of $J(\mathbf{Z}^{(q-1)})$. Furthermore, we assume that $J$ is a tridiagonal matrix and of order $m$. Now we choose $l$ elements from the column vector $\mathbf{Z}^{(q)}$, which we denote by $\mathbf{V}^{(q)}$. This choice is determined by stability considerations (see Section 4). Let $\mathbf{W}^{(q)}$ be the remaining $(m-l)$ elements. Then system (2.6) can be reordered to

$$M \begin{bmatrix} \mathbf{V}^{(q)} \\ \mathbf{W}^{(q)} \end{bmatrix} = P\,\mathbf{B}\,, \tag{2.7}$$

where

$$\begin{bmatrix} \mathbf{V}^{(q)} \\ \mathbf{W}^{(q)} \end{bmatrix} = P\,\mathbf{Z}^{(q)} \text{ and } M = P(I - \theta\, \Delta t\, J)P^T, \tag{2.8}$$

with $P$ a permutation matrix and $P^T$ the transposed matrix $P$. Now we reduce system (2.7), by eliminating $\mathbf{W}^{(q)}$, to a system which only involves $\mathbf{V}^{(q)}$. This can be described by a premultiplication of a matrix $R$. Let $M$ be partitioned according to the separation of $\mathbf{Z}^{(q)}$, i.e.

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

where $M_{11}$ and $M_{22}$ are square matrices. Then $R$ is of the form

$$R = \begin{bmatrix} I & -M_{12}M_{22}^{-1} \\ O & R_{22} \end{bmatrix}, \tag{2.9}$$

where $O$ is a nil matrix. An obvious choice for $R_{22}$ is $R_{22} = M_{22}^{-1}$. This choice is used in the solution method in Appendix B. However, for the incomplete cyclic reduction algorithm (see Appendix A) $R_{22}$ is such that the submatrix $L$ occurring below is a lower triangular matrix :

$$RM = \begin{bmatrix} T & O \\ E & L \end{bmatrix}, \tag{2.10}$$

where

$$T = M_{11} - M_{12}M_{22}^{-1}M_{21}\,,\ E = R_{22}M_{21}\,,\ L = R_{22}M_{22}\,.$$

Here, $T$ is a tridiagonal matrix. Now, system (2.7) becomes of the form

$$\begin{bmatrix} T & O \\ E & L \end{bmatrix} \begin{bmatrix} \mathbf{V}^{(q)} \\ \mathbf{W}^{(q)} \end{bmatrix} = R\,P\,\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}. \tag{2.11}$$

The subsystem

$$T\,\mathbf{V}^{(q)} = \mathbf{B}_1 \tag{2.12}$$

is called the reduced system of equations. Because $L$ is a lower triangular matrix, the elements of $\mathbf{W}^{(q)}$ can be solved straightforwardly once $\mathbf{V}^{(q)}$ is known. In Section 3, we approximate the solution of the reduced system (2.12) by an explicit expression.

EXAMPLE 2.1. Let the indices of the grid points corresponding to the reduced system be given by the set

$$\{j \mid j = l \cdot 2^k, l = 1, \ldots, 2^{p-k} - 1\}, \tag{2.13}$$

where the number of grid points is $2^p - 1$. Then, as said in the introduction, the reduced system appears naturally when $k$ steps of the cyclic reduction algorithm (see Appendix A) are performed on (2.6).

## 3. Approximation of the solution for the reduced system

Let us again consider the difference scheme (2.6), which can be written in the form

$$(I - \theta \Delta t\, J)\mathbf{Z}^{(q)} = (I - \theta \Delta t\, J)\mathbf{Z}^{(0)} + \mathbf{A}, \tag{3.1}$$

where

$$\mathbf{A} = \Delta t\, \{\theta J \mathbf{Z}^{(0)} + \theta \mathbf{g}(\mathbf{Z}^{(q-1)}, t^{n+1}) + (1-\theta)\mathbf{G}(\mathbf{U}^n, \mathbf{U}^n, t^n)\},$$

and $J = J(\mathbf{Z}^{(q-1)})$. Application of the reduction technique of the previous section yields

$$RMP\mathbf{Z}^{(q)} = RMP\mathbf{Z}^{(0)} + R\mathbf{A}.$$

Using (2.8) and (2.10), we may write

$$T\mathbf{V}^{(q)} = T\mathbf{V}^{(0)} + \tilde{\mathbf{B}}_1 \tag{3.2}$$

$$E\mathbf{V}^{(q)} + L\mathbf{W}^{(q)} = E\mathbf{V}^{(0)} + L\mathbf{W}^{(0)} + \tilde{\mathbf{B}}_2$$

where

$$\begin{bmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{B}}_2 \end{bmatrix} = R\mathbf{A}.$$

Now the reduced system (3.2), by which $\mathbf{V}^{(q)}$ is implicitly given, will be approximated by an explicit expression. Therefore we assume that

$$T = D + C. \tag{3.3}$$

The precise form of $C$ ( and consequently of $D$ ) will be given later. For this moment we assume that

$$\rho(D) > \rho(C),$$

where $\rho$ denotes the spectral radius (maximal modulus of the eigenvalues) and that $D^{-1}$ exists and can be computed at low costs. Splittings such as in (3.3) are commonly applied for the construction of iterative methods. Rewriting (3.2), gives

$$\mathbf{V}^{(q)} - \mathbf{V}^{(0)} = (I + D^{-1}C)^{-1}D^{-1}\tilde{\mathbf{B}}_1. \tag{3.4}$$

Using the truncated Neumann series

$$(1 + D^{-1}C)^{-1} \approx 1 - D^{-1}C,$$

we obtain

$$\mathbf{V}^{(q)} - \mathbf{V}^{(0)} \approx (I - D^{-1}C)D^{-1}\tilde{\mathbf{B}}_1.$$

Now, the formula

$$\tilde{\mathbf{V}}^{(q)} = \mathbf{V}^{(0)} + (I - D^{-1}C)D^{-1}\tilde{\mathbf{B}}_1, \tag{3.5}$$

can be used to compute an approximation for the solution of the reduced system.
Approximating the inverse of a matrix, truncated Neumann series are commonly applied with respect to iterative algorithms on vector computers (see [1,2,4,19]).

$$\tilde{\mathbf{V}}^{(q)} = (I - D^{-1}C)D^{-1}\mathbf{B}_1 + (D-1C)^2\mathbf{V}^{(0)} . \tag{3.6}$$

This expression follows by combining (2.12), (3.2) and (3.5). The expression (3.6) is actually used in our numerical experiments.

For the error due to the approximation, we have

$$\|\mathbf{V}^{(q)} - \tilde{\mathbf{V}}^{(q)}\| = \|((I + D^{-1}C)^{-1} - (I - D^{-1}C))\, D^{-1}\tilde{\mathbf{B}}_1\|$$

$$= \Delta t\, \| \frac{(D^{-1}C)^2}{(I + D^{-1}C)} D^{-1}R\,\mathbf{A}\| \leq \Delta t\, \|D^{-1}C\|^2\, \|(I + D^{-1}C)^{-1}\, D^{-1}R\,\mathbf{A}\| .$$

This error is small when $\|D^{-1}C\|$ is small.

The choice of $C$ (and consequently of $D$) is determined by the following considerations :
1. $D$ should be easily invertible, e.g. a diagonal matrix.
2. The replacement of (3.4) by (3.5) should not disturb a possible numerical conservation property of (3.1).
For a discussion of conservation properties of numerical schemes, we refer to [13,16]. In our case, the second consideration is similar to the requirement that the difference

$$\sum_{i=1}^{m}(Z_i^{(q)} - U_i^n) = \mathbf{e}^T(\mathbf{Z}^{(q)} - \mathbf{U}^n) , \quad \text{where} \quad \mathbf{e}^T = [1,1,...,1] , \tag{3.7}$$

is not changed when (3.4) is replaced by (3.5). The difference (3.7) can be evaluated using (3.1). This requirement can be satisfied by choosing

$$D_{ii} = \sum_{j=1}^{l} T_{ij}, \quad D_{ij} = 0 \text{ for } j \neq i , \tag{3.8}$$

and consequently $C = T - D$. By this choice, we have that

$$\mathbf{e}^T C = \mathbf{0}^T , \tag{3.9}$$

where $\mathbf{0}$ is a zero vector. This property does not necessarily imply that a possible conservation property of (3.1) is not disturbed. Hence, we have to calculate the perturbation of (3.1), introduced by the replacement of (3.4) by (3.5), explicitly. Comparison of (3.4) and (3.5), it follows that instead of (3.2) we have solved

$$(T + H)\, \mathbf{V}^{(q)} = (T + H)\, \mathbf{V}^{(0)} + \tilde{\mathbf{B}}_1 , \tag{3.10}$$

where

$$H = C(D^{-1}C)\,(I - D^{-1}C)^{-1} . \tag{3.11}$$

Hence, system (3.10 and 3.11) is identical to (3.5). Since $\mathbf{e}^T C = \mathbf{0}^T$ (see 3.9), we have $\mathbf{e}^T H = \mathbf{0}^T$ . Furthermore, from the definition of $R$ in (2.9), it is easily verified that

$$R^{-1}\begin{bmatrix} H & O \\ O & O \end{bmatrix} = \begin{bmatrix} H & O \\ O & O \end{bmatrix} ,$$

where $O$ denotes a nil matrix. Hence, the modification of (3.1) is given by

$$P^T \begin{bmatrix} H & O \\ O & O \end{bmatrix} P\,(\mathbf{Z}^{(q)} - \mathbf{Z}^{(0)}) .$$

As $\mathbf{e}^T P^T = \mathbf{e}^T$, we have that

$$\mathbf{e}^T P^T \begin{bmatrix} H & O \\ O & O \end{bmatrix} P\,(\mathbf{Z}^{(q)} - \mathbf{Z}^{(0)}) = 0 ,$$

which proves our assertion.

The matrix $D$ can also be seen to originate from a lumping process on the columns of $T$. Lumping is often used in finite element methods (see [18,14]) in order to obtain a diagonal matrix. Furthermore, it is used in the context of multigrid methods[3].

**Summarizing**, the method proceeds as follows :
(a) The system of equations (2.6) is reduced to system (2.11).
(b) $D$ and $C$ are constructed as denoted by (3.3) and (3.8).
(c) The explicit expression (see 3.6) is used to approximate the solution for the reduced system.
(d) $\mathbf{W}^{(q)}$ is solved by back substitution.

REMARK 3.1. In terms of iterative methods for tridiagonal systems, the approximation (3.5) can be considered as one step of the point Jacobi method (see [13],p.138).
This can be explained as follows : If we multiply formula (3.4) with $(I - D^{-1}C)$, we may write

$$(I - (D^{-1}C)^2)\,\mathbf{V}^{(q)} = (I - (D^{-1}C)^2)\,\mathbf{V}^{(0)} + (I - D^{-1}C)D^{-1}\tilde{\mathbf{B}}_1 \,,$$

Now applying one step of the point Jacobi method, where we use $\mathbf{V}^{(0)}$ as an initial approximation for $\mathbf{V}^{(q)}$, gives

$$\begin{aligned}\mathbf{V}^{(q)} &= (I - (D^{-1}C)^2)\,\mathbf{V}^{(0)} + (I - D^{-1}C)D^{-1}\tilde{\mathbf{B}}_1 + (D^{-1}C)^2\,\mathbf{V}^{(0)} .\\ &= \mathbf{V}^{(0)} + (I - D^{-1}C)D^{-1}\tilde{\mathbf{B}}_1 \,,\end{aligned}$$

which corresponds with formula (3.5).

## 1. STABILITY

In this section, a stability condition will be derived for the system of equations (2.11), where the reduced system of equations (2.12) is approximated by scheme (3.5). We only consider linear systems. For the treatment of linear stability theory we refer to [16]. Here, we require that $\|\mathbf{U}^{n+1}\| \leq \|\mathbf{U}^n\|$ for the homogeneous problem, i.e. (2.11) without forcing terms.

The next theorem is used to derive a stability condition for system (2.11).

THEOREM 4.1. *Let $S$ and $J$ be matrices, where $\lambda_S$ and $\lambda_J$ are the corresponding eigenvalues. Then necessary conditions for stability of the scheme*

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t\, S(I - \theta\Delta t J)^{-1} J\, \mathbf{U}^n \,, \quad \tfrac{1}{2} \leq \theta \leq 1 \,, \tag{4.1}$$

*are :*
*(a)* $\mathrm{Re}(\lambda_J) \leq 0$ ,
*(b)* $\lambda_S \in [0,1]$ *and real.*
*Sufficient conditions for stability of this scheme are the conditions (a) and (b), and*
*(c) $S$ and $J$ are normal matrices and commute with each other.*

PROOF. Being commutative, $S$ and $J$ have the same eigensystem. Thus, we arrive at the stability condition

$$|1 + \Delta t \lambda_S (1 - \theta\Delta t \lambda_J)^{-1}\lambda_J| \leq 1 \,. \tag{4.2}$$

This condition means that $\lambda_S(1 - \theta\Delta t\lambda_J)^{-1}\lambda_J$ should be in a circle with centre (-1,0) and radius 1. Due to condition (b), (4.2) is satisfied if

$$|1 + \Delta t(1 - \theta\Delta t\lambda_J)^{-1}\lambda_J| = |\frac{1 + (1-\theta)\Delta t\lambda_J}{1 - \theta\Delta t\lambda_J}| \leq 1 .$$

Since (a) and $\frac{1}{2} \leq \theta \leq 1$, this condition is satisfied. □

Now, the system of equations (2.11) will be written in a form as denoted by (4.1). In the linear case (2.8), we may write for system (2.11)

$$\begin{bmatrix} T & O \\ E & L \end{bmatrix} P\mathbf{U}^{n+1} = \begin{bmatrix} T & O \\ E & L \end{bmatrix} P\mathbf{U}^{n} + \begin{bmatrix} T & O \\ E & L \end{bmatrix} P(I - \theta\Delta t J)^{-1}\,\Delta t\, J\mathbf{U}^{n}, \tag{4.3}$$

where the forcing terms are omitted. This equation is solved by premultiplication of

$$\begin{bmatrix} T & O \\ E & L \end{bmatrix}^{-1} = \begin{bmatrix} I & O \\ E & L \end{bmatrix}^{-1} \begin{bmatrix} T^{-1} & O \\ O & I \end{bmatrix} . \tag{4.4}$$

In our case, we approximate $T^{-1}$ in (4.4) by $K = (I - D^{-1}C)D^{-1}$ (see (3.5)). Now, first (4.3) is premultiplied with

$$\begin{bmatrix} K & O \\ O & I \end{bmatrix} ,$$

which gives

$$\begin{bmatrix} KT & O \\ E & L \end{bmatrix} P\mathbf{U}^{n+1} = \begin{bmatrix} KT & O \\ E & L \end{bmatrix} P\mathbf{U}^{n} + \Delta t \begin{bmatrix} K & O \\ O & I \end{bmatrix} \begin{bmatrix} T & O \\ E & L \end{bmatrix} P(I - \theta\Delta t J)^{-1} J\mathbf{U}^{n} . \tag{4.5}$$

Thereafter, $KT$ is replaced by $I$ and (4.5) is premultiplied by (cf. Remark 3.1)

$$\begin{bmatrix} I & O \\ E & L \end{bmatrix}^{-1} ,$$

which gives the explicit expression

$$P\mathbf{U}^{n+1} = P\mathbf{U}^{n} + \Delta t \begin{bmatrix} I & O \\ E & L \end{bmatrix}^{-1} \begin{bmatrix} K & O \\ O & I \end{bmatrix} \begin{bmatrix} T & O \\ O & I \end{bmatrix} \begin{bmatrix} I & O \\ E & L \end{bmatrix} P(I - \theta\Delta t J)^{-1} J\mathbf{U}^{n} .$$

Finally, this leads to

$$\mathbf{U}^{n+1} = \mathbf{U}^{n+1} + \Delta t P^{T} \begin{bmatrix} I & O \\ E & L \end{bmatrix}^{-1} \begin{bmatrix} KT & O \\ O & I \end{bmatrix} \begin{bmatrix} I & O \\ E & L \end{bmatrix} P(I - \theta\Delta t J)^{-1} J\mathbf{U}^{n} ,$$

Hence $S$ is of the form

$$S = P^{T} \begin{bmatrix} I & O \\ E & L \end{bmatrix}^{-1} \begin{bmatrix} KT & O \\ O & I \end{bmatrix} \begin{bmatrix} I & O \\ E & L \end{bmatrix} P . \tag{4.6}$$

The non-trivial eigenvalues of $S$ are the eigenvalues of $KT$. Then by virtue of Theorem 4.1., the system (2.11) is stable if

$$\lambda_{KT} = \lambda_{\{I - (D^{-1}C)^2\}} \in [0,1] \text{ and real} , \tag{4.7}$$

where the conditions (a) and (b) should be satisfied.

However, condition (c) is not satisfied for $S$ given in (4.6). Since

$$\begin{bmatrix} I & O \\ E & L \end{bmatrix}^{-1} = \begin{bmatrix} I & O \\ -F & L^{-1} \end{bmatrix},$$

where $F = L^{-1}E$, we find for (4.6)

$$S = P^T \begin{bmatrix} KT & O \\ F(I-KT) & I \end{bmatrix} P .$$

The matrix $S$ should commute with $P^T MP$. It is straightforward that among others this leads to the condition

$$KTM_{12} = M_{12} .$$

As in general $M_{12}$ is not a nil matrix, $KT$ should be the identity matrix. This is in general not true. Despite of the fact that sufficient condition (c) is not satisfied, we found in the experiments that the stability condition (4.7) was valid.

REMARK 4.1. The eigenvalues $\lambda_{KT}$ are determined by the choice of the grid points of the reduced system. For a particular choice of $\Delta t$ the magnitude of the norm of $D^{-1}C$ will rapidly tend to zero when the distance between the two nearest points of the explicitly solved system increases. This follows from the fact that the influence of the solution at one point on the solution at other points decreases when the distance between these two points increases. This holds for hyperbolic as well as parabolic problems. For model problems, using the cyclic reduction process, we can derive stability conditions (see (5.4) and (5.7)). The stability conditions are of the form

$$\frac{\Delta t}{(2^k \Delta x)} \leqslant c_{\text{hyp}} ,$$

$$\frac{\Delta t}{(2^k \Delta x)^2} \leqslant c_{\text{parab}} ,$$

where $c_{\text{hyp}}$ and $c_{\text{parab}}$ are constants (the so-called stability boundaries), for hyperbolic and parabolic problems, respectively. Furthermore, $k$ denotes the number of steps in the cyclic reduction process.

REMARK 4.2. Due to the choice of $C$ (see 3.3 and 3.8), we have that $C = 0$ for problems of the form

$$\frac{d}{dt}\mathbf{U} = \Lambda \mathbf{U} ,$$

where $\Lambda$ is a diagonal matrix. Hence for such problems the constructed method is unconditionally stable. This result also holds when $T$ is a diagonal matrix.

REMARK 4.3. The requirement that $KT$ should have real eigenvalues does in general only hold if one starts with central differences for hyperbolic as well as for parabolic problems (see examples). However, in practice often one-sided differences are used. In Section 5.1 we have tested a one-sided difference in the space discretization. again the method gave stable results, although (4.7) is not satisfied in this case.

REMARK 4.4. In the case of a symmetric or antisymmetric Jacobian matrix $J$ with constant coefficients (as below in (5.2) and (5.6)) the matrix $T$ (see (2.10)) is symmetric after one or more steps of the cyclic reduction process. This can be established by performing some steps of the process by hand. Furthermore, the diagonal elements are positive, say $b$ ($b > 0$) and the off-diagonal elements are negative, say $a$ ($a < 0$). Application of the approximation described in Section 3 yields a method which satisfies stability condition (4.7) when $a / b < 1/6$.

REMARK 4.5. From the stability conditions given in Remark 4.1, we observe that the minimal value of $k$, in order to satisfy the stability condition involved, decreases with $\Delta t$ for a constant $\Delta x$. Hence, $\Delta t$ determines the number of reduction steps and thus the needed implicitness. Therefore, for a given time step $\Delta t$, the computation time is minimal, when we use the minimal value of $k$ such that the stability condition involved, is satisfied.

## 5. Numerical illustration

To illustrate the performance of the method described in Sections 2 and 3, we present some experiments, both for linear and nonlinear problems. In the experiments the cyclic reduction algorithm is used to solve the equations. By varying the set of points which are solved explicitly (see (2.8)), we vary the stability property of the method. Our choice will be the regular set of grid points as denoted by (2.13), where k denotes the number of cyclic reduction steps. In this case, we have

$$\mathbf{V} = [\ U_{2^k}, U_{2\cdot 2^k}, U_{3\cdot 2^k}, \ldots, U_{2^p-2^k}\ ]^T ,$$

where the superscripts are omitted. To make optimal use of the cyclic reduction algorithm, we have chosen a uniform grid such that the number of grid points is $N = 2^p - 1$, although this is not essential, with mesh size $\Delta x = L / 2^p$. The aim of our experiments is to show the relation between the number of reduction steps, which is a measure for the implicitness, and the stability behaviour of the applied method. Furthermore, we are interested in the accuracy behaviour when the number of reduction steps varies. To measure the obtained accuracy we define

$$\text{cd} = -{}^{10}\log(\ |\ \text{maximal global error at the endpoint } t = T\ |\ ),$$

denoting the number of correct digits in the numerical approximation at the endpoint. The calculations were performed on the CDC Cyber 170-750 which has a 48-bit mantissa, i.e. a machine precision of about 14 decimal digits.

### *5.1. A linear hyperbolic problem*

As a first example, consider the linear test problem

$$u_t = u_x\ ,\quad 0<t<T,\quad 0<x<L\ , \tag{5.1}$$

with initial condition

$$u(x,0) = \sin(2\pi x / L)\ ,$$

and boundary condition

$$u(L,t) = \sin(2\pi(L+t)/L)\ .$$

The exact solution is given by

$$u(x,t) = \sin(2\pi(x+t)/L)\ ,$$

where $L = 64$ .

Central differences are used at all points except for the first point where a commonly used one-sided difference is applied. In the notation of the split function G (see 2.4) the discretization is given by

$$\begin{aligned}
(J\mathbf{U})_1 &= \frac{(U_2 - U_1)}{\Delta x}\ , \quad g_1 = 0\ , \\
(J\mathbf{U})_j &= \frac{(U_{j+1} - U_{j-1})}{2\Delta x}\ , \quad g_j = 0 \ \text{ for } j = 2,\ldots,N-1\ , \\
(J\mathbf{U})_N &= \frac{-\ U_{N-1}}{2\Delta x}\ , \quad g_N(t) = \sin(2\pi(L+t)/L)/(2\Delta x)\ .
\end{aligned} \tag{5.2}$$

Notice that for linear hyperbolic systems, the Jacobian matrix $J$ has almost purely imaginary eigenvalues. For the time integration, we used the Trapezoidal Rule, $\theta = \frac{1}{2}$ ($Q = 1$ due to linearity).

Only for linear test problems we compare our approach with incomplete cyclic reduction without any adaptation (see Introduction). For nonlinear test problems we expect the same behaviour. In the case of incomplete cyclic reduction without any adaptation, the reduced system of equations is solved by (cf. 3.5)

$$\tilde{\mathbf{V}}^{(q)} = (I + D^{-1}C)\mathbf{V}^{(0)} + D^{-1}\tilde{\mathbf{B}}_1 \ , \tag{5.3}$$

where

$$D_{ii} = T_{ii} \ , \ D_{ij} = 0 \ \text{ for } \ j \neq i \ .$$

In Table 5.1 we give the cd-values of the method, obtained at the endpoint $T = 320$. In the last column we listed the values for the incomplete cyclic reduction method.

| | scheme (3.5) | | | | | | scheme (5.3) |
|---|---|---|---|---|---|---|---|
| $\Delta t$ | $p = 5$<br>$\Delta x = 2$ | $p = 6$<br>$\Delta x = 1$ | $p = 7$<br>$\Delta x = 0.5$ | $p = 8$<br>$\Delta x = 0.25$ | $p = 9$<br>$\Delta x = 0.125$ | | $p = 8$<br>$\Delta x = 0.25$ |
| 1 | 1.30(1)<br>1.30(2) | 1.80(1)<br>1.80(2) | 1.87(1)<br>2.11(2) | ***(1)<br>2.24(2)<br>2.25(3) | ***(2)<br>2.27(3) | | ***(2)<br>0.34(3)<br>2.03(4)<br>2.25(5) |
| 2 | 1.19(1)<br>1.19(2) | 1.24(1)<br>1.51(2)<br>1.51(3) | ***(1)<br>1.63(2)<br>1.64(3) | ***(2)<br>1.67(3) | ***(3)<br>1.68(4) | | ***(3)<br>0.64(4)<br>1.68(5) |
| 4 | 0.78(1)<br>0.91(2)<br>0.91(3) | ***(1)<br>1.01(2)<br>1.04(3)<br>1.04(4) | ***(2)<br>1.05(3)<br>1.08(4) | ***(3)<br>1.06(4) | | | ***(4)<br>0.88(5)<br>1.08(6) |
| 8 | ***(1)<br>0.36(2)<br>0.43(3)<br>0.43(4) | ***(2)<br>0.40(3)<br>0.46(4) | ***(3)<br>0.41(4) | | | | |

Table 5.1. Number of correct digits for the linear hyperbolic problem (5.1) with $T = 320$.

In Table 5.1 the number of cyclic reduction steps is given in parenthesis. An unstable behaviour of the integration process is dented by ***. The number of grid points is equal to $2^p$.

The results clearly show the effect of varying the number of reduction steps:
(a) The error hardly depends on the number of reduction steps, as long as the computation is stable.
(b) If the mesh size is decreased by a factor two then one extra reduction step is needed to maintain the same stability boundary on $\Delta t$.

For scheme (5.3) at least two extra reduction steps are needed to obtain accuracy which is comparable with scheme (3.5). It can be proved that scheme (5.3) is unstable. This scheme is only useful when the off-diagonal elements are neglectable with respect to the diagonal elements.

From (4.7) we can derive, by performing some reduction steps explicitly, the stability condition

$$\frac{\Delta t}{(2^k \Delta x)} \leqslant c_k \,, \quad k = 1,\dots,p-1 \,, \tag{5.4}$$

where $k$ denotes the number of cyclic reduction steps and $c_k$ is a constant depending on $k$. If $k$ equals $p$ then the method is purely implicit and unconditionally stable. In Table 5.2 we have listed the values $c_k$, for $k = 1,\dots,6$.

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $c_k$ | 1 | 1.09868 | 1.12546 | 1.13230 | 1.13402 | 1.13445 |

Table 5.2. Stability coefficients for hyperbolic problem (5.1).

It appeared that

$$\lim_{k\to\infty} c_k \uparrow 1.134593 \,.$$

REMARK 5.1. From the results in Table 5.1 it is easily verified that condition (5.4) is satisfied. As the number of grid points, at the old time level, involved in the computation of the solution at the new time level is of $O(2^k)$ after $k$ reduction steps, we have from the stability condition (5.4) that the time step increases almost linearly with the size of the influence domain.

REMARK 5.2. Notice that we did not test the method with zero cyclic reduction steps. In this case, it can be shown that $D^{-1}C = 0.5\,\Delta t\, J$ with respect to the internal points. Applying immediately the explicit expression (3.6) we obtain a second-order method which has similar properties as the two-stage second-order Runge-Kutta method[9]. The latter is not stable for problems from which the Jacobian matrix has imaginary eigenvalues. After one step of the cyclic reduction algorithm the eigenvalues of the resulting matrix $T$ (see 2.10)) are real when we start with a Jacobian matrix $J$ with imaginary eigenvalues. In this case, the matrix $C$ given by (3.3) and (3.8) has imaginary eigenvalues and thereby the eigenvalues of $KT$ (see (4.7)) are not real.

We also tested our method for the same linear hyperbolic problem, using one-sided differences for $u_x$. In this case, $J$ and g are is given by

$$(J\mathbf{U})_j = \frac{(U_{j+1} - U_j)}{\Delta x} \,, \quad \mathrm{g}_j = 0 \ \text{ for } j = 1,\dots,N-1 \,,$$

$$(J\mathbf{U})_N = \frac{-\,U_N}{\Delta x} \,, \quad \mathrm{g}_N(t) = \sin(2\pi(L+t)/L)/\Delta x \,.$$

In this case the eigenvalues of $KT$ (see (4.7)) are not real. Notwithstanding, this scheme shows a comparable stability behaviour as in the case of central differences (see 5.2). The results are given in the same form as in Table 5.1.

| $\Delta t$ | $p = 5$<br>$\Delta x = 2$ | $p = 6$<br>$\Delta x = 1$ | $p = 7$<br>$\Delta x = 0.5$ | $p = 8$<br>$\Delta x = 0.25$ | $p = 9$<br>$\Delta x = 0.125$ |
|---|---|---|---|---|---|
| 1 | 0.43(0)<br>0.44(1)<br>0.44(2) | 0.66(0)<br>0.68(1)<br>0.68(2) | ***(0)<br>0.95(1)<br>0.95(2) | ***(1)<br>1.23(2)<br>1.23(3) | ***(2)<br>1.46(3) |
| 2 | 0.42(0)<br>0.43(1)<br>0.43(2) | ***(0)<br>0.67(1)<br>0.68(2)<br>0.68(3) | ***(1)<br>0.94(2)<br>0.94(3) | ***(2)<br>1.20(3)<br>1.21(4) | ***(3)<br>1.45(4) |
| 4 | 0.41(1)<br>0.42(2)<br>0.42(3) | ***(1)<br>0.64(2)<br>0.65(3)<br>0.65(4) | ***(2)<br>0.86(3)<br>0.86(4) | ***(3)<br>1.03(4) | |
| 8 | ***(1)<br>0.34(2)<br>0.35(3)<br>0.35(4) | ***(2)<br>0.46(3)<br>0.46(4) | ***(3)<br>0.53(4) | | |

Table 5.3. Number of correct digits for the linear hyperbolic problem (5.1) with one-sided differences and $T = 320$.

## *5.2. A linear parabolic problem*

As a second example, consider the linear test problem

$$\begin{aligned} &u_t = u_{xx}\,, \quad 0<t<T, \quad 0<x<L\,, \\ &u_x(0,t) = \frac{2\pi}{L}\, e^{-(\frac{2\pi}{L})^2 t} \\ &u(L,t) = 0\,. \end{aligned} \tag{5.5}$$

The exact solution is given by

$$u(x,t) = e^{-(\frac{2\pi}{L})^2 t} \sin(2\pi x\,/\,L)\,,$$

where $L = 32$ .

For the space-discretization of (5.5), central differences are used which yields for $J$ and $\mathbf{g}$ (see 2.4)

$$
\begin{aligned}
(J\mathbf{U})_1 &= \frac{(U_2 - U_1)}{(\Delta x)^2}, \quad g_1(t) = -(\frac{2\pi}{L} e^{-(\frac{2\pi}{L})^2 t})/\Delta x\,, \\
(J\mathbf{U})_j &= \frac{(U_{j-1} - 2U_j + U_{j+1})}{(\Delta x)^2}, \quad g_j(t) = 0 \text{ for } j = 2,\ldots,N-1\,, \\
(J\mathbf{U})_N &= \frac{(U_{N-1} - 2U_N)}{(\Delta x)^2}, \quad g_N(t) = 0\,.
\end{aligned} \tag{5.6}
$$

Here, $x_j = x_0 + j\Delta x$ with $x_0 = -\frac{1}{2}\Delta x$. Furthermore, $\Delta x$ should be such that $x_{N+1} = L$.
The Jacobian matrix $J$, given by (5.6), has real eigenvalues. For the time integration we used the Trapezoidal Rule, i.e. $\theta = 0.5$ ($Q = 1$ due to linearity).

The results are given in the same form as in Table 5.1.

| | scheme (3.5) | | | | | scheme (5.3) |
|---|---|---|---|---|---|---|
| $\Delta t$ | $p=4$<br>$\Delta x=2$ | $p=5$<br>$\Delta x=1$ | $p=6$<br>$\Delta x=0.5$ | $p=7$<br>$\Delta x=0.25$ | | $p=7$<br>$\Delta x=0.25$ |
| 2 | 2.00(0)<br>2.13(1) | ***(0)<br>2.64(1)<br>2.69(2) | ***(1)<br>3.00(2)<br>3.41(3)<br>3.43(4) | ***(2)<br>3.40(3)<br>4.07(4) | | ***(4)<br>3.97(5) |
| 4 | 2.12(1)<br>2.16(2) | ***(1)<br>2.75(2)<br>2.85(3) | ***(2)<br>3.49(3)<br>3.37(4) | ***(3)<br>3.31(4) | | ***(4)<br>3.12(5) |
| 8 | 2.30(2) | 2.28(2)<br>2.78(3)<br>2.76(4) | ***(2)<br>2.38(3)<br>2.56(4) | ***(3)<br>2.40(4) | | ***(4)<br>2.50(5) |
| 16 | 2.24(2)<br>2.16(3) | ***(2)<br>2.01(3)<br>1.97(4) | ***(3)<br>1.94(4)<br>1.94(5) | ***(4)<br>1.91(5) | | ***(4)<br>1.90(5) |

Table 5.4. Number of correct digits for the linear parabolic problem (5.5) with $T = 32$.

Globally, we observe the same effect for this parabolic problem as for the hyperbolic problem (5.3). If the mesh size is decreased by a factor four, then one extra reduction step is needed to maintain the same stability boundary on $\Delta t$. Here the numerical error is not only determined by the time integration. For small time steps, compared with the space mesh, the space discretization error becomes visible.

As for the hyperbolic problem, some extra reduction steps are needed for scheme (5.3) in order to obtain accuracy which is comparable with scheme (3.5).

From (4.7) we obtained the stability condition (cf. (5.4))

$$\frac{\Delta t}{(2^k\,\Delta x)^2} \leqslant c_k\ ,\quad k = 0,\ldots,p-1\ . \tag{5.7}$$

In Table 5.5 we have listed some values for $c_k$.

| $k$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $c_k$ | 0.5 | 0.60355 | 0.63334 | 0.64105 | 0.64299 |

Table 5.5. Stability coefficients for parabolic problem (5.5).

Here, we have

$$\lim_{k\to\infty} c_k \uparrow 0.643651\ .$$

It is easily verified that condition (5.7) is in agreement with the results in Table 5.4. The results for this parabolic problem show a similar behaviour as for the hyperbolic problem (5.1). For every applied reduction step the maximal time step increases with about a factor four. Thus the maximal time step increases almost quadratically with the size of the influence domain for the difference equation.

## *5.3. A nonlinear parabolic problem*

Consider the nonlinear one-dimensional heat equation (see [15])

$$\frac{\partial u}{\partial t} = \frac{1}{\rho c(u)}\frac{\partial}{\partial x}\,(K(u)\frac{\partial u}{\partial x})\ , \tag{5.8}$$

with $u$ the temperature, $\rho c$ the heat capacity and $K$ the thermal conductivity. The thermal conductivity and the heat capacity are given by

$$K(u) = 1 + 0.5u$$

$$\rho c(u) = 1 + 0.5u\ .$$

We consider a finite bar, with boundary conditions

$$u_x(0,t) = -\frac{1}{K(u(0,t))}$$

$$u_x(L,t) = 0\ ,$$

where $L = 2$. So the bar is isolated at the endpoint $L$ and at $x=0$ a constant heat input $q = -K(u)\frac{\partial u}{\partial x} = 1$ is assumed. Due to the nonlinear nature of equation (5.8), we have

$$\mathbf{G}(\tilde{\mathbf{U}},\mathbf{U},t) = A(\mathbf{K}(\tilde{\mathbf{U}}))\mathbf{U} + \mathbf{g}(\tilde{\mathbf{U}},t)\ ,$$

where

$$(A(\mathbf{K}(\tilde{\mathbf{U}}))\mathbf{U})_j = \frac{1}{(\rho c(\tilde{\mathbf{U}}))_j} \frac{((\mathbf{K}(\tilde{\mathbf{U}})D_x\mathbf{U})_{j+\frac{1}{2}} - (\mathbf{K}(\tilde{\mathbf{U}})D_x\mathbf{U})_{j-\frac{1}{2}})}{\Delta x}. \tag{5.9}$$

In (5.9) $\mathbf{K}(\tilde{\mathbf{U}})$ is given by

$$(\mathbf{K}(\tilde{\mathbf{U}}))_{j+\frac{1}{2}} = 1 + 0.5\frac{(\tilde{U}_{j+1} + \tilde{U}_j)}{2},$$

and

$$(\rho c(\tilde{\mathbf{U}}))_j = 1 + 0.5\,\tilde{U}_j\,.$$

Furthermore, $D_x\mathbf{U}$ and $\mathbf{g}$ are given by

$$(D_x\mathbf{U})_{\frac{1}{2}} = 0, \quad \mathbf{g}_1(\tilde{\mathbf{U}},t) = \frac{-1}{(\rho c(\tilde{\mathbf{U}}))_1} / \Delta x\,,$$

$$(D_x\mathbf{U})_{j+\frac{1}{2}} = \frac{(U_{j+1} - U_j)}{\Delta x}, \quad \mathbf{g}_j(\tilde{\mathbf{U}},t) = 0 \ \text{ for } j = 2,\dots,N-1,$$

$$(D_x\mathbf{U})_{N+\frac{1}{2}} = 0, \quad \mathbf{g}_N(\tilde{\mathbf{U}},t) = 0.$$

Hence, the matrix $A$ is tridiagonal. The grid points are chosen $x_j = j\Delta x + x_0$ with $x_0 = -\frac{1}{2}\Delta x$. The mesh width $\Delta x$ should be such that $x_{N+\frac{1}{2}} = L$.
For the time integration we used (2.6) with $Q = 2$ ($\theta$ will be given later).
Furthermore, for $q = 1$ $\mathbf{Z}^{(q)}$ is solved using $k-1$ reduction steps in the cyclic reduction algorithm, and for $q = 2$ $\mathbf{Z}^{(q)}$ is solved using $k$ reduction steps.
As the first step is a prediction only, we have used there one reduction step less than in the second step. Other choices are, of course, possible.

In the nonlinear experiments, we determined a reference solution using a very small integration step and we only considered the error due to the time integration. We have chosen a space mesh $2/(2^5-1)$. For the time integration we used the Trapezoidal Rule ($\theta = 0.5$) and the Backward Euler method ($\theta = 1$). The results are given in Table 5.6.

| $\Delta t$ | $\theta = 0.5$ | | | $\theta = 1$ | | |
|---|---|---|---|---|---|---|
| | $k=2$ | $k=3$ | $k=4$ | $k=2$ | $k=3$ | $k=4$ |
| 0.0125 | 4.11 | 4.23 | | 2.33 | 2.76 | 2.76 |
| 0.025 | 2.88 | 3.57 | 3.59 | *** | 2.47 | 2.48 |
| 0.05 | *** | 2.73 | 2.73 | | 2.15 | 2.16 |
| 0.1 | | 2.03 | 2.02 | | 1.79 | 1.87 |
| 0.2 | | 1.34 | 1.34 | | *** | 1.61 |

Table 5.6. Number of correct digits for the nonlinear parabolic problem (5.8) with $T = 1$ and $k$ the number of cyclic reduction steps.

The results of this example give rise to conclusions similar to the previous examples : when the number of cyclic reductions increases, and thus the implicitness, the stability of the method increases.

The results clearly show the first-order behaviour of the Backward Euler method and the second-order behaviour of the Trapezoidal Rule. The maximal time step for the Trapezoidal Rule is twice the maximal step for the Backward Euler method. This is due to the fact that the magnitude of elements of matrix $A(\mathbf{K}(\mathbf{U}))$ of the Backward Euler method are twice the magnitude of the elements of the Trapezoidal Rule.

## *5.4. A nonlinear hyperbolic problem*

Consider a simplified form of the shallow water equations in one dimension :

$$\begin{aligned} u_t &= -\lambda u - g\zeta_x , \\ \zeta_t &= -(hu)_x , \end{aligned} \tag{5.10}$$

where $u$ denotes the depth-averaged velocity, $\lambda$ the bottom friction, $\zeta$ the elevation of the water surface, $\bar{h}$ the depth when the water is in rest, $h$ the total depth given by $h = \bar{h} + \zeta$ and $g$ the acceleration due to gravity. The first equation is a momentum equation describing the change in time of the velocity $u$. The second equation is a continuity equation. The numerical solution of (5.10) is required in the region

$$0 \leqslant x \leqslant L , \text{ for } 0 \leqslant t \leqslant T .$$

The boundary conditions are

$$u(0,t) = -\sin(\omega t) \quad \text{and} \quad \zeta(L,t) = \cos(\omega t) .$$

The initial conditions are given by

$$u(x,0) = 0 \quad \text{and} \quad \zeta(x,0) = 1 .$$

Let $\mathbf{W} = (\mathbf{U},\mathbf{Z})^T$, where $\mathbf{U}_j(t) \approx u(j\Delta x,t)$ and $\mathbf{Z}_j(t) \approx \zeta(j\Delta x,t)$ . In this case $\mathbf{G}(\mathbf{W},\tilde{\mathbf{W}},t)$ is defined by

$$\begin{aligned} \mathbf{G}^U(\mathbf{W},\tilde{\mathbf{W}},t) &= -gD_x\tilde{\mathbf{Z}} - \lambda\tilde{\mathbf{U}} . \\ \mathbf{G}^Z(\mathbf{W},\tilde{\mathbf{W}},t) &= -(M\mathbf{H}\, D_x\tilde{\mathbf{U}} + M\mathbf{U}\, D_x\tilde{\mathbf{H}}) . \end{aligned} \tag{5.11}$$

where

$$\begin{aligned} (D_x\mathbf{U})_j &= (U_{j+1} - U_{j-1})/\Delta x , \\ H_j &= \bar{h} + \frac{(Z_{j-1} + Z_{j+1})}{2} , \\ (M\mathbf{H})_j &= \frac{(H_{j-1} + H_{j+1})}{2} . \end{aligned}$$

At the grid point on the left boundary we used

$$H_1 = \bar{h} + Z_2 .$$

In this case, we will not give the precise form of $J$ and $\mathbf{g}$ (see 2.4), but they can be derived straightforwardly from (5.11). By this choice for the space-discretization it can be shown that two independent sets of equations arise, i.e. a system for $(U_{2k},Z_{2k+1})$, and a system for $((U_{2k+1},Z_{2k+2})$, $k=1,2,3,...)$. Hence, by omitting the latter system the number of variables is reduced by a factor two. The variables of the remaining system are now space staggered. For more details on space staggering we refer to [5,17].

The variant of (2.5) which will be used here, is the following method :

$$\mathbf{W}^{(0)} = \mathbf{W}^n$$

$$\mathbf{W}^{(q)} = \mathbf{W}^n + \frac{\Delta t}{2}\{\mathbf{G}(\mathbf{W}^{(q-1)},\mathbf{W}^{(q)},t^{n+1}) + \mathbf{G}(\mathbf{W}^n,\mathbf{W}^n,t^n)\} \quad \text{for } q=1,...,Q \tag{5.12}$$

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \frac{\Delta t}{2}\{\mathbf{G}(\mathbf{W}^{(Q)},\mathbf{W}^{(Q)},t^{n+1}) + \mathbf{G}(\mathbf{W}^n,\mathbf{W}^n,t^n)\}\,.$$

The first $Q$ steps give each rise to a linear implicit equation for $\mathbf{W}^{(q)}$, whereas the last step is purely explicit. For $Q \geqslant 1$ this method is second-order accurate in time. Using this method possible conservation properties of the semi-discretization are preserved irrespective of the linearization used in the second equation. In our case, the second equation in (5.12) represents mass conservation. This conservation is only simulated by $\mathbf{G}^Z$ if the first and second argument of $\mathbf{G}^Z$ are equal, because in this case it holds that

$$\mathbf{G}^Z(\mathbf{W},\mathbf{W},t) = -(M\mathbf{H}\, D_x\mathbf{U} + M\mathbf{U}\, D_x\mathbf{H}) = -D_x(\mathbf{HU})\,. \tag{5.13}$$

Hence, the second equation of (5.12) does not simulate this conservation property as long as $\mathbf{W}^{(q)} \neq \mathbf{W}^{(q-1)}$. However, the third equation makes the overall method conservative for all choices of $Q$.

If the expression for $\mathbf{U}^{(q)}$ is substituted into the equation for $\mathbf{Z}^{(q)}$, then the second equation from (5.12) gives rise to a tridiagonal system for $\mathbf{Z}^{(q)}$. Once $\mathbf{Z}^{(q)}$ is solved, $\mathbf{U}^{(q)}$ can be solved straightforwardly.

In Table 5.7 results are given for the case $Q = 1$. The constants are chosen

$$T = 5120\ ,\ \Delta x = 25\ ,\ L = 3175\ ,\ \omega = \frac{2\pi}{3600}\ ,\ g = 9.81 \text{ and } \bar{h} = 10\,.$$

| $\Delta t$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|
| 10 | 4.71 | 4.71 | 4.71 | 4.71 |
| 20 | *** | 4.00 | 4.01 | 4.01 |
| 40 | | *** | 3.37 | 3.38 |
| 80 | | | *** | 2.01 |

Table 5.7. Number of correct digits for the nonlinear hyperbolic problem (5.10) with $k$ the number of cyclic reduction steps.

These results show again that the error does not depend on the number of reduction steps, as long as the computation is stable. As in the linear hyperbolic case, about a factor two is gained when an extra reduction step is applied. When $k$ equals five, the method is unconditionally stable. If we perform the first step (see (5.12)) twice, we obtain comparable results, except when ($\Delta t = 80$ and $k = 5$). In that case, we have a considerable increase of the number of correct digits.

## 6. Conclusions

In this paper we have constructed explicit-implicit methods starting from a one-step implicit method, which yields a tridiagonal system. These methods were constructed for time dependent partial differential equations. The method makes use of the fact that the interdependence of the solution at two different points at the new time level decreases if the physical distance between these points increases. This fact causes that the magnitude of the off-diagonals decrease rapidly when compared with the main diagonal in each step of the cyclic reduction process. The constructed methods have the following properties :

(a) The accuracy is hardly influenced if we replace the one-step implicit method by an approximating explicit-implicit method as long as the integration is stable.

(b) If the one-step implicit method satisfies a conservation property, then this property is preserved when the implicit method is replaced by the approximating explicit-implicit method.

(c) The maximum allowed time step increases linearly or quadratically with the number of points of the old time level which influence the solution at a point at the new time level for hyperbolic or parabolic equations, respectively.

In this paper, we have restricted ourselves to one-dimensional problems. However, the technique can be applied directly to alternating direction methods often used in multi-dimensional cases. Such methods lead to a succession of one-dimensional problems, each of which can be treated by the described technique.
The approximation of fully implicit methods in the multi-dimensional case by explicit-implicit methods is subject of future research. We expect that the theory for this case will develop along the same lines.

## 7. References

[1] Adams, L., m-Step preconditioned conjugate gradient methods, *SIAM J. Sci. Stat. Comput.*, vol. 6 (1985), pp. 452-463.

[2] Axelsson, O., A survey of preconditioned iterative methods for linear systems of equations, *BIT*, vol. 25 (1985), pp. 166-187.

[3] Dendy, J.K. jr., Black box multigrid for nonsymmetric problems, *Appl. Math. and Comput.*, vol 13 (1983), pp. 261-283.

[4] Dubois, P.P., Greenbaum, A. and G.H. Rodrigue, Approximating the inverse matrix for use in iterative algorithms on vector computers, *Computing*, vol. 22 (1979), pp. 257-268.

[5] Hansen, W. Theorie zur errechnung das wasserstandes und der stromungen in randmeeren nebst anwendungen, *Tellus*, vol. 8 (1956), pp. 287-300.

[6] Heller, D., Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems, *SIAM J. Numer. Anal.*, vol. 13 (1976), pp. 484-496.

[7] Hockney, R.W., A fast direct solution of Poisson's equation using Fourier analysis, *J. Assoc. Comp. Mach.*, vol. 12 (1965), pp. 95-113.

[8] Hockney, R.W. and C.R. Jesshope, *Parallel computers : architecture, programming and algorithms*, Adam Hilger, Ltd., Bristol, 1981.

[9] Houwen, P.J., *Construction of integration formulas for initial-value problems*, North-Holland, Amsterdam, 1977.

[10] Houwen, P.J. and J.G. Verwer, One-step splitting methods for semi-discrete parabolic equations, *Computing*, vol 22 (1979), pp.291-309.

[11] Lambert, J.D., *Computational methods in ordinary differential equations*, Wiley, London-New York, 1973.

[12] Lambiotte, J.J. and R.G. Voight, The solution of tridiagonal linear systems on the CDC Star-100 computer, *ACM Trans. Math. Software*, vol. 1 (1975), pp. 308-329.

[13] Mitchell, A.R. and D.F. Griffiths, *The finite difference method in partial differential*

*equations*, Wiley, Chichester, 1980.

[14] MITCHELL, A.R. AND R.W. WAIT, *The finite element analysis and applications*, John Wiley, New York, 1985.

[15] ORIVUORI, S., Efficient method for solution of non-linear heat conduction problems, *Int. J. for Num. Meth. in Eng.*, vol. 14 (1979), pp. 1461-1476.

[16] RICHTMYER, R.D. AND K.W. MORTON, *Difference methods for initial value problems*, Wiley, New York, 1967.

[17] STELLING, G., *On the construction of computational methods for shallow water flow problems*, Thesis TH Delft, 1983.

[18] STRANG, G. AND J. FIX, *An analysis of the finite element method*, Prentice Hall, Englewood Cliffs, 1973.

[19] VORST, H.A. VAN DER, A vectorizable variant of some ICCG methods, *SIAM J. Sci. Stat. Comput.*, vol. 3 (1982), pp.350-356.

[20] VORST, H.A. VAN DER, *Vectorization of linear recurrence relations*, Faculty of Mathematics & Informatics Delft, in preparation.

[21] WANG, H.H., A parallel method for tridiagonal system equations, *ACM Trans. on Math. Softw.*, vol. 7 (1981), pp. 170-183.

## APPENDIX A

In Appendix A and B we give two possible solution methods for system (2.7), which yield a system of equations as denoted by (2.11).

Matrix decomposition I : Cyclic reduction

The cyclic reduction algorithm was originally developed by Hockney[7], for the discrete version of Poisson's equation. The cyclic reduction algorithm is well-suited for use on a parallel or vector computer, as many of the quantities involved may be computed independently of the others. This case has been studied by Lambiotte and Voight [12], with attention to a vector computer.

We assume that the system of linear algebraic equations arising from implicit difference formula (2.3), which must be solved at each time step is a special case of the tridiagonal system

$$\alpha_j x_{j-1} + \beta_j x_j + \gamma_j x_{j+1} = b_j ,$$

for $1 \leq j \leq m$ , where $\alpha_1 = 0$ and $\gamma_m = 0$.
Also, we assume that $m = 2^p - 1$, although this is not essential, where $p$ is some positive integer. In matrix form, we obtain

$$\begin{bmatrix} \beta_1 & \gamma_1 & & & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & & \\ & \cdot & \cdot & \cdot & \\ & & \alpha_{m-1} & \beta_{m-1} & \gamma_{m-1} \\ 0 & & & \alpha_m & \beta_m \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_{m-1} \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ b_{m-1} \\ b_m \end{bmatrix} . \tag{A.1}$$

The cyclic reduction algorithm separates the system in two subsystems, which involve respectively the rows with even indices and the rows with odd indices. Let us rewrite (A.1) as follows :

$$\begin{array}{lllll} \alpha_{j-1} x_{j-2} + \beta_{j-1} x_{j-1} + \gamma_{j-1} x_j & & & = b_{j-1} \\ & \alpha_j x_{j-1} + \beta_j x_j + \gamma_j x_{j+1} & & = b_j \ . \\ & & \alpha_{j+1} x_j + \beta_{j+1} x_{j+1} + \gamma_{j+1} x_{j+2} & = b_{j+1} \end{array}$$

Multiplying the first equation by $-\alpha_j / \beta_{j-1}$, the third by $-\gamma_j / \beta_{j+1}$ and adding to the second equation, we obtain

$$\left(\frac{-\alpha_{j-1}\alpha_j}{\beta_{j-1}}\right) x_{j-2} + \left(\beta_j - \frac{\gamma_{j-1}\alpha_j}{\beta_{j-1}} - \frac{\alpha_{j+1}\gamma_j}{\beta_{j+1}}\right) x_j + \left(\frac{-\gamma_{j+1}\gamma_j}{\beta_{j+1}}\right) x_{j+2} = \tag{A.2}$$

$$\frac{-\alpha_j}{\beta_{j-1}} b_{j-1} + b_j + \frac{-\gamma_j}{\beta_{j+1}} b_{j+1} .$$

In order to simplify the notation, we introduce

$$\kappa_j = \left(\frac{-\alpha_{j-1}\alpha_j}{\beta_{j-1}}\right), \lambda_j = \left(\beta_j - \frac{\gamma_{j-1}\alpha_j}{\beta_{j-1}} - \frac{\alpha_{j+1}\gamma_j}{\beta_{j+1}}\right) , \mu_j = \left(\frac{-\gamma_{j+1}\gamma_j}{\beta_{j+1}}\right) \text{ and}$$

$$\frac{-\alpha_j}{\beta_{j-1}} b_{j-1} + b_j + \frac{-\gamma_j}{\beta_{j+1}} b_{j+1} = B_j .$$

Then (A.2) is equal to

$$\kappa_j x_{j-2} + \lambda_j x_j + \mu_j x_{j+2} = B_j .$$

Thus, if j is even, the new system of equations involves $x_j$'s with even indices. Similar equations hold for $x_2$ and $x_{m-1}$. The process of reducing the equations in this fashion is known as cyclic reduction. Then (A.1) may be written as the following equivalent system :

$$\begin{bmatrix} \lambda_2 & \mu_2 & & & 0 \\ \kappa_4 & \lambda_4 & \mu_4 & & \\ & \cdot & \cdot & \cdot & \\ & & \kappa_{m-3} & \lambda_{m-3} & \mu_{m-3} \\ 0 & & & \kappa_{m-1} & \lambda_{m-1} \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ \cdot \\ x_{m-3} \\ x_{m-1} \end{bmatrix} = \begin{bmatrix} B_2 \\ B_4 \\ \cdot \\ B_{m-3} \\ B_{m-1} \end{bmatrix} . \tag{A.3}$$

and

$$\begin{bmatrix} \beta_1 & 0 & & & 0 \\ 0 & \beta_3 & 0 & & \\ & \cdot & \cdot & \cdot & \\ & & 0 & \beta_{m-2} & 0 \\ 0 & & & 0 & \beta_m \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \cdot \\ x_{m-2} \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_3 \\ \cdot \\ b_{m-2} \\ b_m \end{bmatrix} - \begin{bmatrix} \gamma_1 & & & 0 \\ \alpha_3 & \gamma_3 & & \\ & \cdot & \cdot & \\ & & \alpha_{m-2} & \gamma_{m-2} \\ 0 & & & \alpha_m \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ \cdot \\ x_{m-3} \\ x_{m-1} \end{bmatrix} . \tag{A.4}$$

Since $m = 2^p - 1$ and the new system (A.3) involves only $x_j$'s with even indices, the dimension of the new system is $2^{p-1} - 1$. Note that once (A.3) is solved, it is easy to solve for the $x_j$'s with odd indices, as evidenced by (A.4). The system (A.4) is known as the eliminated equations.

Since system (A.3) is tridiagonal and in the form of (A.1), we can apply the reduction algorithm repeatedly until we have one equation. However, we can stop the process after any step and use another method to solve the reduced system of equations. After renumbering, we obtain a system of equations as denoted by (2.11).

## Appendix B

Matrix decomposition II : A parallel method.
Here, we use a variant on Wang's algorithm [21]. Let us assume that the system of linear equations given in (A.1), is of the form

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{d}_1 & & & 0 \\ \mathbf{a}_k & \beta_k & \mathbf{c}_k & & \\ & \mathbf{e}_2 & \mathbf{A}_2 & \mathbf{d}_2 & \\ & & \mathbf{a}_l & \beta_l & \mathbf{c}_l \\ 0 & & & \mathbf{e}_3 & \mathbf{A}_3 \end{bmatrix} \mathbf{x} = \mathbf{b} \,, \tag{B.1}$$

where $\mathbf{A}_1$,$\mathbf{A}_2$ and $\mathbf{A}_3$ are tridiagonal matrices and

$$\mathbf{a}_i = [0,...,0,\alpha_i] \;,\; \mathbf{c}_i = [\gamma_i,0,...,0] \;,\; \mathbf{d}_1 = [0,...,0,\gamma_{k-1}]^T \,,$$
$$\mathbf{d}_2 = [0,...,0,\gamma_{l-1}]^T \;,\; \mathbf{e}_2 = [\alpha_{k+1},0,...,0]^T \;,\; \mathbf{e}_3 = [\alpha_{l+1},0,...,0]^T \,.$$

In this example, we use three block matrices, but this reduction technique can be applied for an arbitrary number of block matrices. For this subdivision $x_k$ and $x_l$ will be the unknowns of the reduced system of equations. If the block matrices are invertible, then system (B.1) may be replaced by

$$\begin{bmatrix} \mathbf{A}_1^{-1} & & & & 0 \\ & 1 & & & \\ & & \mathbf{A}_2^{-1} & & \\ & & & 1 & \\ 0 & & & & \mathbf{A}_3^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{d}_1 & & & 0 \\ \mathbf{a}_k & \beta_k & \mathbf{c}_k & & \\ & \mathbf{e}_2 & \mathbf{A}_2 & \mathbf{d}_2 & \\ & & \mathbf{a}_l & \beta_l & \mathbf{c}_l \\ 0 & & & \mathbf{e}_3 & \mathbf{A}_3 \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{A}_1^{-1} & & & & 0 \\ & 1 & & & \\ & & \mathbf{A}_2^{-1} & & \\ & & & 1 & \\ 0 & & & & \mathbf{A}_3^{-1} \end{bmatrix} \mathbf{b} \,,$$

which is equivalent to

$$\begin{bmatrix} \mathbf{I} & \mathbf{v}_1 & & & 0 \\ \mathbf{a}_k & \beta_k & \mathbf{c}_k & & \\ & \mathbf{w}_2 & \mathbf{I} & \mathbf{v}_2 & \\ & & \mathbf{a}_l & \beta_l & \mathbf{c}_l \\ 0 & & & \mathbf{w}_3 & \mathbf{I} \end{bmatrix} \mathbf{x} = \mathbf{b}' \,,$$

where the $\mathbf{v}$ and $\mathbf{w}$ are column vectors with $\mathbf{v}_i = \mathbf{A}_i^{-1}\mathbf{d}_i$ and $\mathbf{w}_i = \mathbf{A}_i^{-1}\mathbf{e}_i$. So far, this method corresponds with the first steps of Wang's algorithm. Now, we eliminate $\mathbf{a}_k$ , $\mathbf{c}_k$ , $\mathbf{a}_l$ and $\mathbf{c}_l$, which yields

$$\begin{bmatrix} \mathbf{I} & \mathbf{v}_1 & & & 0 \\ & \beta_k' & & \nu_k & \\ & \mathbf{w}_2 & \mathbf{I} & \mathbf{v}_2 & \\ & \nu_l & & \beta_l' & \\ 0 & & & \mathbf{w}_3 & \mathbf{I} \end{bmatrix} \mathbf{x} = \mathbf{b}'' \,.$$

By a simple reordering this system can be brought to a system of the form (2.11). The $k^{\text{th}}$ and the $l^{\text{th}}$ row, which do not contain elements of the block matrices, form the reduced system of equations. It should be noted that the elimination of the off-diagonal elements of the matrices $\mathbf{A}_i$ can de done independently. Thereby, this approach is well suited for vector and parallel computers (see [20]).