stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 136/82 SEPTEMBER

H.B. DE VRIES

THE TWO-LEVEL ALGORITHM IN THE SOLUTION OF THE INITIAL VALUE PROBLEM FOR PARTIAL DIFFERENTIAL EQUATIONS

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics subject classification: 65B99, 65F10, 65M20

The two-level algorithm in the solution of the initial value problem for partial differential equations

by

H.B. de Vries

ABSTRACT

An implicit linear multistep method is applied to semi-discretized (nonlinear) time-dependent partial differential equations. Using Newton iteration the nonlinear implicit relations are replaced by a sequence of linear equations. These linear equations are solved by the iterative use of a two-level algorithm. An iteration method based on incomplete LU-decomposition is chosen as relaxation method (smoothing operator) in the two-level algorithm. Using discrete Fourier analysis the convergence and smoothing properties of the relaxation method are investigated for a model problem. The numerical experiments illustrate the choice of some suitable operators and parameters in the two-level algorithm.

KEY WORDS & PHRASES: *Numerical analysis, method of lines, initial-boundary value problems, multi-grid methods, incomplete LU-decomposition*

1. INTRODUCTION

Consider a system of ODE's of the form

$$(1.1) \qquad \frac{d^{\nu}y}{dt^{\nu}} = f(t,y) \quad , \ \nu = 1,2,$$

obtained from the semi-discretization of a time-dependent 2-dimensional (non-linear) hyperbolic (ν=2) or parabolic (ν=1) partial differential equation (PDE). Suppose that we decide to integrate this initial value problem by an implicit linear multistep method. This leads us to the problem to solve in each integration step the system of equations

$$(1.2) \qquad y - b_0\tau^{\nu}f(t_{n+1},y) = \sum_{\ell=1}^{k} [a_{\ell}y_{n+1-\ell} + b_{\ell}\tau^{\nu}f(t_{n+1-\ell},y_{n+1-\ell})],$$

where y_n denotes the numerical solution at $t = t_n$, $\tau = t_{n+1}-t_n$ and $\{a_{\ell},b_{\ell}\}$ are real coefficients. The (approximate) solution of this equation is identified with y_{n+1}.

In [6] the *PCGC-method* (Preconditioning and Coarse Grid Corrections) has been described to solve the system (1.2). In view of the implementation we will present and formulate this method in a different manner (see section 2), which shows also more clearly the inner and outer iteration in the method and the resemblance with *two-level algorithms* as described in [2].

In section 3 we discuss three possible choices of *incomplete LU-decomposition* (ILU-decomposition) and give some arguments why the ILU-decomposition) defined in [6] has been chosen.

The computational work of the PCGC-method based on arithmetic operations is considered in section 4.

Finally, in section 5 we apply the PCGC-method to two parabolic PDE's and illustrate several choices of the operators and parameters in the PCGC-method.

2. THE TWO-LEVEL ALGORITHM

In the PCGC method an iteration scheme is defined, which consists of an inner and outer iteration. In our present approach we shall separate these iterations.

Using the *modified Newton-Raphson process* we replace the system of equations (1.2) by a sequence (m) of systems of linear equations:

$$
\begin{aligned}
& y^{(0)} = y^{(\mathrm{pred})} \ , \\
& [I - b_0 \tau^{\nu} J]\, y = \phi^{(j-1)} \quad , \ j = 1,\ldots,m \ , \\
& J = \frac{\partial f}{\partial y}\,(t_{n+1}, y^{(0)}) \ , \\
& \phi^{(j-1)} = \Sigma_n + b_0 \tau^{\nu}\, [f(t_{n+1}, \bar{y}) - J\bar{y}] \ ,
\end{aligned}
\tag{2.1}
$$

where Σ_n denotes the right-hand side of equation (1.2), $y^{(0)}$ is obtained by some predictor formula, $\bar{y}$ is the solution of the preceding system of linear equations with $\bar{y} = y^{(0)}$ for $j = 0$.

In the outer iteration (2.1) each of the systems of linear equations are solved by the iterative use of a *two-level algorithm* [2] (inner iteration). The two-level algorithm uses two computational grids (viz. the fine grid Ω_h with grid parameter h and the coarse grid Ω_H with grid parameter H = 2h) and the corresponding sets of gridfunctions on Ω_h and Ω_H , U^h and U^H, respectively. The convergence of the iteration process for solving on Ω_h

$$
[I - b_0 \tau^{\nu} J]\, y = \phi^{(j-1)} \tag{2.2}
$$

(for each $j \in \{1,\ldots,m\}$) will be accelerated by using *defect corrections*, which are obtained by solving on Ω_H the approximate problem

$$
[I - b_0 \tau^{\nu} J_H]\, y_H = \psi_H^{(j)} \ , \tag{2.3}
$$

where H refers to the grid Ω_H. The Jacobian matrix J_H and $\psi_H^{(j)}$ will be defined later.

Before we describe the two-level algorithm (TLA) we introduce the *restriction operator* R_{Hh} and the *prolongation operator* P_{hH}:

(2.4a) $R_{Hh}: U^h \to U^H$,

(2.4b) $P_{hH}: U^H \to U^h$.

The *injection* I_{Hh} (or 1-point restriction [9])

(2.4c) $I_{Hh}: U^h \to U^H$

copies only in the corresponding grid points of Ω_h and Ω_H the function values of the grid function on Ω_h to a grid function on Ω_H. In the numerical experiments we used the weighted restriction R_{Hh} and the linear interpolation P_{hH} as defined in [6].

The Jacobian matrix J_H is defined by

$$(2.5) \qquad J_H = \frac{\partial f_H}{\partial y_H}(t_{n+1}, y_H^{(0)}) ,$$

where $y_H = I_{Hh} y$, $y_H^{(0)} = I_{Hh} y^{(0)}$ and f_H denotes the right-hand side function of (1.1) corresponding to Ω_H.

Finally, we introduce an iteration method based on *incomplete LU-decomposition*. The matrix $I - b_0 \tau^\nu J$ is decomposed (see section 3) as follows:

$$(2.6) \qquad I - b_0 \tau^\nu J = L^* U^* - R ,$$

where R is the residual matrix and L^*, U^* are a lower and upper triangular matrix, respectively.

The iterative method based on (2.6) reads

$$(2.7) \qquad L^* U^* \, x_{i+1} = R x_i + \phi^{(j-1)} .$$

In section 3 three forms of this ILU-relaxation method will be considered.

In the two-level algorithm the problem (2.3) on Ω_H will not be solved directly, but *iteratively* by means of an iteration method based on incomplete LU-decomposition of $I - b_0 \tau^\nu J_H$, i.e.,

(2.8a) $$z_0 = (L_H^* U_H^*)^{-1} \psi_H^{(j)} ,$$

(2.8b) $$L_H^* U_H^* z_{i+1} = \tilde{R}_H z_i + \psi_H^{(j)} , \quad i = 0,\ldots,\rho - 1,$$

where $$L_H^* U_H^* - \tilde{R}_H = [I - b_0 \tau^\nu J_H].$$

One *coarse grid correction* step in the two-level algorithm reads

(2.9a) $$x = x + c_h ,$$

where the correction term c_h [6] is defined by

(2.9b) $$c_h = P_{hH} [I - b_0 \tau^\nu J_H]^{-1} \psi_H^{(j)} ,$$

(2.9c) $$\psi_H^{(j)} = R_{Hh}(\phi^{(j-1)} - [I - b_0 \tau^\nu J] x).$$

On Ω_H the coarse grid problem (2.3) with $\psi_H^{(j)}$ defined by (2.9c) is solved after ρ iterations with (2.8a)-(2.8b).

The two-level algorithm (TLA) can be described in quasi-Algol as follows:

```
proc TLA = (vec y, φ^(j-1)) vec:
begin
    vec x, ψ_H^(j), z;
    x: = y;
    for i to p do x: = (L*U*)^-1 [R x + φ^(j-1)] od ;
    ψ_H^(j) := R_Hh(φ^(j-1) - [ I - b_0 τ^ν J] x);
    z: = (L_H* U_H*)^-1 ψ_H^(j);
    for ℓ to ρ - 1 do z:= (L_H* U_H*)^-1 [R̃_H z + ψ_H^(j)] od ;
    x:= x + P_hH z ;
    for i to s do x:= (L*U*)^-1 [R x + φ^(j-1)] od ;
    x
end;
```

One step in the two-level algorithm TLA consists of p ILU-relaxations (2.7), a coarse grid correction step (2.9a)-(2.9c) and again s ILU-relaxations (2.7). In order to describe exactly what particular TLA algorithm is

used we introduce the following notations (cf.[6]):

I iteration step defined by (2.7)

C_ρ coarse grid correction defined by (2.9b)-(2.9c), where ρ is the number of iterations with (2.8a)-(2.8b) on the coarse grid Ω_H.

One two-level iteration, i.e. one execution of $TLA(y,\phi^{(j-1)})$, is now denoted by

$$I^p \, C_\rho \, I^s \quad . \tag{2.10}$$

Let the evaluation of $I-b_0 \tau^\nu J$ and $I-b_0 \tau^\nu J_H$ be defined by

EJAC ≡ GENJAC (t,h,y),

EJACH ≡ GENJAC (t,H,I_{Hh}y) ,

respectively, and let the ILU-decomposition of these matrices be defined by

LUR ≡ ICLUDEC (EJAC,w) ,

LUH ≡ ICLUDEC (EJACH,w),

respectively, where w is a parameter which selects the particular ILU-decomposition to be chosen. In case of constant partial derivatives $\partial f/\partial y$ the matrices $I - b_0 \tau^\nu J$ and $I - b_0 \tau^\nu J_H$ and their ILU-decompositions are determined once; in all other cases they were updated every integration step. Then the iteration scheme per integration step τ for nonlinear PDE's, consisting of the outer iteration (Newton) and the inner iteration (TLA algorithm), can now be written in quasi-Algol as follows:

```
t := t + τ;  y := y(pred);
EJAC := GENJAC (t,h,y);
EJACH := GENJAC (t,H,IHhy);
LUR := ICLUDEC(EJAC,w);
LUH := ICLUDEC (EJACH,w);
```

for j *to* m *do* # outer iteration #

$\phi^{(j-1)} := \Sigma_n + b_0 \tau^\nu [f(t,y) + EJAC * y - y]$;

for i *to* k *do* # inner iteration #

y := TLA ($\phi^{(j-1)}$, y)

od

od;

In order to describe the PCGC method [6] we introduce the notations:

- E evaluation of the function $\phi^{(j-1)}$ defined in (2.1)
- m number of right-hand side evaluations $\phi^{(j-1)}$ per integration step
- k number of TLA iterations per Newton step
- r number of coarse grid corrections (2.9a)-(2.9c) per integration step
- M number of fine grid iterations (2.7) per integration step.

A particular PCGC method with coarse grid corrections is now denoted by

$$(2.11) \qquad [E(I^p C_\rho I^s)^k]^m \quad \text{with } k = \frac{r}{m} .$$

REMARK 2.1. Performing no coarse grid corrections (2.9a)-(2.9c), i.e. r = 0, the method is denoted by (see [6])

$$(2.11') \qquad (E\, I^{M/m})^m .$$

REMARK 2.2. When the ILU-relaxation (2.7) is used the residual $\phi^{(j-1)} - (I-b_0\tau^\nu J)x_i$ can be computed as follows

$$\phi^{(j-1)} - (I-b_0\tau^\nu J)x_i := L^* U^* x_i - R x_{i-1} - L^* U^* x_i + R x_i :=$$

$$:= R\,(x_i - x_{i-1}) ,$$

where x_{i-1} and x_i are the (i-1) - and i - th iterand obtained by (2.7). Since usually the error matrix R has less non-zero diagonals than $I - b_0\tau^\nu J$, this

is a very cheap way to compute the residual. For p = 1 in TLA the residual occurring in $\psi_H^{(j)}$ will be computed as follows:

$$\phi^{(j-1)} - (I - b_0 \tau^\nu J)\, x := R\,(x-y)\ ,$$

without using an extra array storage.

The amplification matrix M_{ILU} of the ILU-relaxation (2.7) on Ω_h is given by

$$M_{ILU} = (L^*U^*)^{-1} R = I - (L^*U^*)^{-1} A_h\ , \tag{2.13}$$

where $A_h = I - b_0 \tau^\nu J$. When in the coarse grid correction step (2.9) the system of linear equations (2.3) is solved exactly on Ω_H, then the amplification matrix of (2.9) (cf. HEMKER [2]) M_{CGC} is given by

$$M_{CGC} = [I - P_{hH} A_H^{-1} R_{Hh} A_h]\ , \tag{2.14}$$

where $A_H = I - b_0 \tau^\nu J_H$. From a straightforward calculation it follows that the ρ - th iterand of (2.8) is given by

$$z_\rho = \sum_{i=1}^{\rho-1} [(L_H^* U_H^*)^{-1} \tilde{R}_H]^i\, L_H^* U_H^*\ \psi_H^{(j)} =$$

$$= (\ I\ - [(L_H^* U_H^*)^{-1} \tilde{R}_H]^\rho)\ A_H^{-1}\ \psi_H^{(j)}\ ,$$

where $\psi_H^{(j)} = R_{Hh}\,(\phi^{(j-1)} - A_h x)$.
Then it can be easily shown, that the amplification matrix $\tilde{M}_{CGC}$ of (2.9), where in the evaluation of the correction term (2.9b)-(2.9c) the equation (2.3) is solved after ρ iterations with (2.8a)-(2.8b) is given by

$$\tilde{M}_{CGC} = [I - P_{hH}[I - ((L_H^* U_H^*)^{-1} \tilde{R}_H)^\rho]\, A_H^{-1}\, R_{Hh}\, A_h] = \tag{2.15}$$

$$= M_{CGC} + P_{hH}\,((L_H^* U_H^*)^{-1}\, \tilde{R}_H)^\rho\, A_H^{-1}\, R_{Hh}\, A_h\ .$$

If ρ is sufficiently large, we can expect that $\|\tilde{M}_{CGC}\| \cong \|M_{CGC}\|$ where $\|\ \|$ is some suitable norm.

From (2.13) and (2.15) it follows that the amplification matrix M_{TLA} of one step of the two-level algorithm TLA is given by (cf.HEMKER [2])

$$M_{TLA} = [(L^*U^*)^{-1}R]^s \; \tilde{M}_{CGC}[(L^*U^*)^{-1}R]^p . \tag{2.16}$$

3. THE INCOMPLETE LU-DECOMPOSITION

3.1. Three possible choices of ILU-decompositions

Let J be a (K×K)-matrix, then writing $A_h = I - b_0\tau^\nu J$ and denoting the elements of the matrices A_h, L^*, U^* and R by a_{ij}, ℓ^*_{ij}, u^*_{ij} and r_{ij}, $1 \le i,j \le K$, the ILU-decomposition (2.6) is defined by

$$\begin{aligned}
&\ell^*_{jj} = 1 \;, \; j = 1,\ldots,K,\\
&\text{If } (k,j) \in P \text{ then } u^*_{kj} = 0 \Rightarrow r_{kj} := -\Big(a_{kj} - \sum_{i=1}^{k-1} \ell^*_{ki} u^*_{ij}\Big)\\
&\qquad \text{else } u^*_{kj} := a_{kj} - \sum_{i=1}^{k-1} \ell^*_{ki} u^*_{ij} \text{ for } j = k,\ldots,K;\\
&\text{If } (j,k) \in P \text{ then } \ell^*_{jk} = 0 \Rightarrow r_{jk} := -\Big(a_{jk} - \sum_{i=1}^{k-1} \ell^*_{ji} u^*_{ik}\Big)\\
&\qquad \text{else } \ell^*_{jk} := \Big(a_{jk} - \sum_{i=1}^{k-1} \ell^*_{ji} u^*_{ik}\Big)/u^*_{kk} \text{ for } j = k+1,\ldots,K,
\end{aligned} \tag{3.1}$$

where $k = 1,\ldots, K$ and P is a set of pairs of integers, which is a subset of $P_K \equiv \{(i,j) \mid i \ne j,\ 1 \le i \le K,\ \ 1 \le j \le K\}$.

When the PDE does not contain mixed derivatives and is semi-discretized by standard finite differences the components in f are coupled according to the familiar five-point molecule.In this case the matrix A_h has a 5-diagonal form and the places (i,j) with $a_{ij} = 0$ are given by the set

$$P_0 = \{(i,j) \mid |i-j| \ne 0,1,b \;\; ; \; 1 \le i,j \le K\}, \tag{3.2}$$

where b is the half-bandwidth of A_h; i.e., $a_{ij} = 0$ whenever $|i-j| > b$. The matrix A_h is schematically shown in figure 3.1.

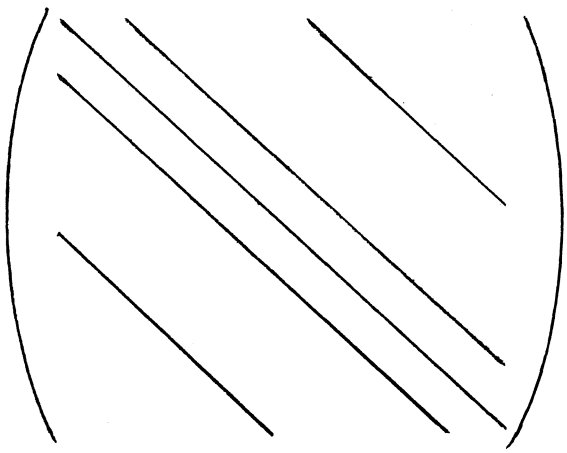

Fig. 3.1. Form of matrix A_h

Several well-known iterative methods (2.7) can be obtained by properly choosing $P \subset P_K$ [7] :

$P = P_K$ results in the point Jacobi method,

$P = \{(i,j) \mid i < j\}$ results in the point Gauss-Seidel method.

The ILU-decomposition based on

$$P_{b-2} \equiv \{(i,j) \mid |i-j| \neq x, x = 0(1)b; 1 \leq i,j \leq K\} = \{(i,j) \mid |i-j| > b; 1 \leq i,j \leq K\}$$

results in the (complete) LU-decomposition of A_h.

Here we will consider three choices for P, which are suitable for 5-diagonal matrices (see fig. 3.1).

The first choice is the set P_0 defined by (3.2). The ILU-decomposition based on P_0 is called in [9] ILU-5.

The second choice, already given in [6], is the set of pairs of integers defined by

$$P_1 = \{(i,j) \mid |i-j| \neq 0,1,b-1,b \ ; \ 1 \leq i,j \leq K\}. \tag{3.3}$$

The ILU-decomposition based on P_1 is called in [9] ILU-7.

The third choice is the set of pairs of integers defined by

(3.4) $$P_2 = \{(i,j) \mid |i-j| \neq 0,1,b-2,b-1,b;\ 1 \leq i,j \leq K\}.$$

The ILU-9 decomposition is based on the set P_2.

REMARK 3.1. In the description of the iteration scheme in section 2 we have defined the ILU-decomposition of A_h and A_H by

LUR ≡ ICLUDEC(EJAC,w) and LUH ≡ ICLUDEC(EJACH,w), respectively.

The ILU-5, ILU-7 and ILU-9 decomposition can be obtained by choosing w equal to 0,1 and 2, respectively. The (complete) LU-decomposition of A_h can be obtained by choosing w equal to b - 2.

For future reference we determine of each ILU-decomposition and relaxation sweep the number of arithmetic operations on a uniform grid Ω_h with N inner points (see also [9]). An operation will be defined as an element from the set { +, -, * , /, sqrt}. In table 3.1 we give the number of operations on a uniform grid with N inner points to perform the decomposition and one relaxation step (2.7) for the three choices of P, viz. (3.2), (3.3) and (3.4). The number of arrays of length N required for storage of the matrices occurring in ILU-5, ILU-7 and ILU-9 are also given in table 3.1.

Table 3.1. Number of arithmetic operations and the number of storage arrays of length N required for ILU-5, ILU-7 and ILU-9 on a uniform grid with N inner points.

Method	Number of operations of the decomposition	Number of operations of the ILU-step (2.7)	Number of storage arrays
ILU - 5	8 N	13 N	7
ILU - 7	17 N	17 N	9
ILU - 9	28 N	25 N	13

3.2. Convergence and smoothing-factors of the three ILU-relaxation methods

Consider the forms of L^*, U^* and R schematically shown in figure 3.2.

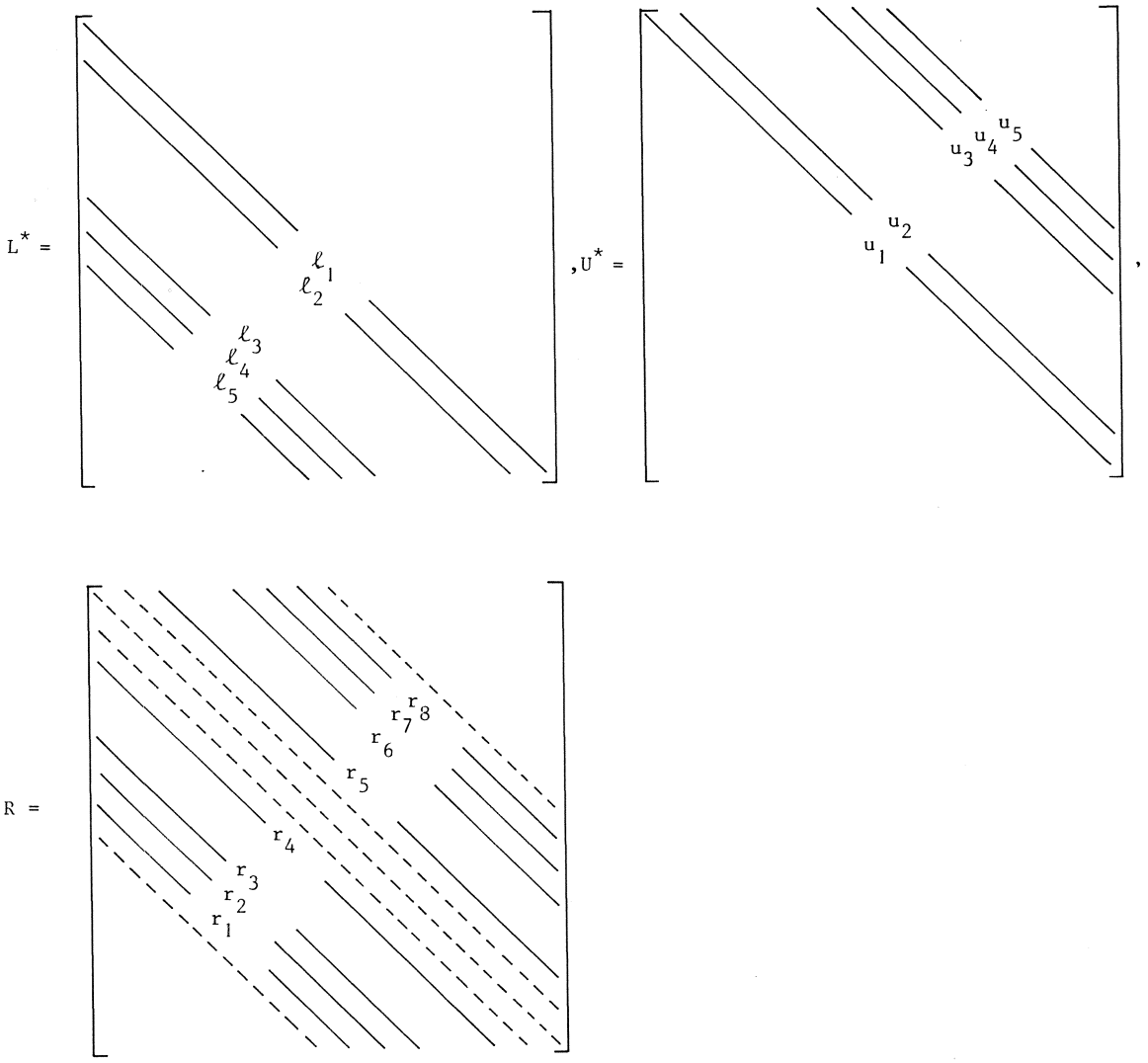

Fig. 3.2. The forms of L^*, U^* and R

In figure 3.2 we have denoted the possible non-zero diagonals in L^* by ℓ_i, $i = 1,\ldots,5$, in U^* by u_i, $i = 1,\ldots,5$, and in R by r_i, $i = 1,\ldots,8$.

For convenience we give below the non-zero diagonals occurring in L^*, U^* and R for each ILU-decomposition:

1^o ILU - 5 with P_o (3.2): r_1, r_8 and ℓ_i, u_i for $i = 1,2,5$.
2^o ILU - 7 with P_1 (3.3): r_2, r_7 and ℓ_i, u_i for $i = 1,2,4,5$.
3^o ILU - 9 with P_2 (3.4): r_i for $i = 3,4,5,6$ and ℓ_i, u_i for $i = 1,2,3,4,5$.

Consider PDE's of the form

$$(3.5) \qquad \frac{\partial^\nu u}{\partial t^\nu} = \alpha(u_{x_1x_1} + u_{x_2x_2}) + v(t, x_1, x_2) \quad , \quad \nu = 1,2 \ , \ \alpha > 0$$

defined on $\Omega = \{(x_1,x_2) \mid 0 \le x_1, x_2 \le 1\}$ and $0 \le t \le 1$ with Dirichlet boundary conditions and initial conditions for u (and $\frac{\partial u}{\partial t}$) at $t = 0$. Let such problems be semi-discretized on a uniform grid Ω_h by standard symmetric differences. In this case the matrix $A_h = I - b_0\tau^\nu J$ has constant elements.

Using a Fourier decomposition of the errors at the internal grid points (see also [1,10]) we can study separately the convergence rate of each Fourier component, and in particular the convergence rate of high-frequency components, which determines the rate of smoothing. In the multi-grid method the role of relaxation is not to reduce the error but to reduce the high-frequency components of the error and the lower frequencies being reduced by relaxation sweeps on coarser grids. For more details on the smoothing analysis we refer to [1,3,4,10].

The Fourier mode analysis is only representative for the local behaviour in the interior of the domain of the PDE, i.e., the boundary conditions are neglected. Therefore we suppose that A_h, L^*, U^* and R are infinite Toeplitz matrices [3,10]. In this case the matrix A_h can be given by the following difference molecule

$$(3.6) \qquad A_h = \begin{bmatrix} & a_1 & \\ a_1 & a_0 & a_1 \\ & a_1 & \end{bmatrix} \quad , \quad a_1 = -\frac{\alpha b_0 \tau^\nu}{h^2} \ , \ a_0 = 1 - 4a_1 .$$

Using the notation of BRANDT [1] we obtain as convergence factor $\mu(\theta)$ with $\theta = (\theta_1, \theta_2)$ for the ILU-5 relaxation:

$$(3.7)\qquad \mu_5(\theta) = \frac{2\tilde{r}_1\cos(\theta_1-\theta_2)}{a_0+2a_1(\cos\theta_1+\cos\theta_2)+2\tilde{r}_1\cos(\theta_1-\theta_2)}\ ,\ r_1 = a_1(-1+\tfrac{1}{2}\sqrt{2})$$

and for the ILU-7 relaxation:

$$(3.8)\qquad \mu_7(\theta) = \frac{2\tilde{r}_2\cos(2\theta_1-\theta_2)}{a_0+2a_1(\cos\theta_1+\cos\theta_2)+2\tilde{r}_2\cos(2\theta_1-\theta_2)}\ ,\ \tilde{r}_2 = a_1(-0.11181)$$

and for the ILU-9 relaxation:

$$(3.9)\qquad \mu_9(\theta) = \frac{2\tilde{r}_3\cos(3\theta_1-\theta_2)+2\tilde{r}_4\cos\,2\theta_1}{a_0+2a_1(\cos\theta_1+\cos\theta_2)+2\tilde{r}_3\cos(3\theta_1-\theta_2)+2\tilde{r}_4\cos\,2\theta_1}\ ,$$

$$\tilde{r}_3 = a_1(-0.03961)\quad ,\ \tilde{r}_4 = a_1(-0.03548).$$

The coefficient $\tilde{r}_1$ can be found analytically, whereas the coefficients $\tilde{r}_2,\tilde{r}_3$ and $\tilde{r}_4$ are numerically derived. For the Poisson equation, i.e. $a_0 = 4$ and $a_1 = -1$, the factors $\mu(\theta)$ of ILU-5 and ILU-7 are already given in [8].

In a multigrid method we are interested in the smoothing factor [1,10], which is defined by

$$(3.10)\qquad \bar{\mu} = \max_{\pi/2 \le |\theta| \le \pi} |\mu(\theta)|\ ,$$

where $\theta = (\theta_1,\theta_2)$ and $|\theta| = \max(|\theta_1|,|\theta_2|)$. For the model problems (3.5)-(3.6) with $h^2 \ll 1$ the smoothing factors $\bar{\mu}_5,\bar{\mu}_7$ and $\bar{\mu}_9$ of ILU-5, ILU-7 and ILU-9, respectively, are listed in table 3.2. In order to compare the efficiency of the ILU-iteration methods we have also listed in table 3.2 the number of operations per ILU-iteration step (n_i, i=5,7,9) and the number of operations per ILU-iteration step for 10^{-1} reduction of the high frequency components of the error ($n_i/|\log \bar{\mu}_i|$, i = 5,7,9) on a uniform grid Ω_h with N inner points.

Table 3.2. Smoothing factors $\bar{\mu}_i$, the number of operations per ILU-step n_i and the number of operations for 10^{-1} reduction of the high frequency components of the error $n_i/|\log \bar{\mu}_i|$ on a uniform grid Ω_h with N inner points for problem (3.5)-(3.6).

Relaxation method	$\bar{\mu}_i$	n_i	$n_i/\|\log \bar{\mu}_i\|$
ILU - 5, i = 5	0. 2035	13N	18.80N
ILU - 7, i = 7	0. 1259	17N	18.89N
ILU - 9, i = 9	0. 052	25N	19.47N

For the class of problems (3.5) ILU-5 has slightly better efficiency if we take only the smoothing of the error into account.

However, in the two-level algorithm defined in section 2 the role of the ILU-relaxation (2.8) on Ω_H is to reduce the error. Therefore we will consider also the low-frequency components of the error.

Assume $-\pi < \theta_i = \omega_i h\pi < \pi$, $\omega_i \in \mathbb{Z}\setminus\{0\}$, i = 1,2 and $\frac{b_0\tau^\nu\alpha}{h^2} \gg 1$, i.e. we assume that $a_0 = 1 - 4a_1 \simeq -4a_1$. Then it can be numerically verified that the convergence factors $\mu_5(\theta)$, $\mu_7(\theta)$ and $\mu_9(\theta)$ reach their maxima $\hat{\mu}_5, \hat{\mu}_7$ and $\hat{\mu}_9$, respectively, in $(\theta_1,\theta_2) = (\pi h,\pi h)$ (see also appendix A). In table 3.3 the convergence factors of the ILU-5, ILU-7 and ILU-9 relaxation are illustrated. On the basis of this table ILU-7 reduces the low frequency component $(\theta_1,\theta_2) = (\pi h,\pi h)$ of the error more efficient than ILU-5. The results listed in the tables 3.1, 3.2 and 3.3 indicate that the ILU-7 relaxation seems to be the best choice in the two-level algorithm for the class of problems (3.5). This is justified by numerical experiments in section 5.3.

REMARK 3.2. Suppose that $a_1 = -\frac{b_0\tau^\nu\alpha}{h^2} \to -\infty$, as $\tau,h \to 0$, then the rate of convergence $\hat{\mu}_i$ of ILU - 5, ILU - 7 and ILU - 9 relaxation is asymptotically

$$\hat{\mu}_5 \simeq 1 - \frac{\pi^2}{(1-\frac{1}{2}\sqrt{2})} h^2 ,$$

$$\hat{\mu}_7 \simeq 1 - \frac{\pi^2}{0.11181} h^2 \quad \text{and}$$

$$\hat{\mu}_9 \simeq 1 - \frac{\pi^2}{0.07509} h^2, \text{ respectively.}$$

Table 3.3. The convergence factor $\hat{\mu}_i \equiv \mu_i(\pi h, \pi h)$ for i = 5,7,9 and the number of operations for 0.1 reduction of the low frequency component $(\theta_1,\theta_2) = (\pi h, \pi h)$ of the error $(n_i/|\log \hat{\mu}_i|$ for i=5,7,9) on Ω_h for problem (3.5)-(3.6) with $\frac{b_0 \tau^{\nu_\alpha}}{h^2} \gg 1$.

h	ILU -5		ILU -7		ILU - 9	
	$\hat{\mu}_5$	$n_5/\|\log \hat{\mu}_5\|$	$\hat{\mu}_7$	$n_7/\|\log \hat{\mu}_7\|$	$\hat{\mu}_9$	$n_9/\|\log \hat{\mu}_9\|$
1/10	0.7495	103.8N	0.5207	60N	0.3829	60N
1/12	0.8112	143.1N	0.6131	80N	0.4883	80.3N
1/16	0.884	242.8N	0.7405	130.3N	0.6435	130.6N
1/20	0.9224	370.6N	0.8177	194.5N	0.7436	194.3N
1/24	0.9448	527.2N	0.8663	272.7N	0.8091	271.7N
1/32	0.9682	926.3N	0.9203	471.3N	0.8844	468.6N
1/40	0.9794	1438.1N	0.9476	727.3N	0.9233	721.4N
1/48	0.9856	2063.7N	0.963	1038.3N	0.9456	1029.1N

4. THE COMPUTATIONAL WORK OF THE PCGC METHOD

In this section an estimate will be derived for the computational work of the PCGC method. An operation will be defined as an element from the set {+,-,*,/,sqrt}. Let the uniform fine grid Ω_h have N inner points, and the uniform coarse grid Ω_H (H=2h) have $N_H \simeq \frac{N}{4}$ inner points.

4.1. The computational work of one TLA-iteration

Here we derive an estimate of the computational work W_{TLA} to perform one TLA-iteration. In table 4.1 several notations are introduced for the

number of operations for the different parts of the algorithm TLA (see section 2).

Table 4.1. Notations for the number of operations for the different parts of TLA in $I^p C_\rho I^s$ mode.

Part of procedure TLA	number of operations on Ω_h
$x := (L^*U^*)^{-1}\,[Rx + \phi^{(j-1)}]$	n_a
$\psi_H^{(j)} := R_{Hh}(\phi^{(j-1)} - [I - b_0\tau^\nu J]x)$	n_b
$z := (L_H^*U_H^*)^{-1}\,\psi_H^{(j)}$	n_c
$z := (L_H^*U_H^*)^{-1}\,[\tilde{R}_H z + \psi_H^{(j)}]$	n_d
$x := x + P_{hH} z$	n_e

The number of operations for one TLA-iteration in $I^p C_\rho I^s$ mode is given by the following formula

$$W_{TLA} = (p+s)n_a + n_b + n_c + (\rho-1)n_d + n_e .$$

The number of operations for the weighted restriction R_{Hh} (*9-point restriction* in [9]) defined in [6] is: 2.75N.
In the injection I_{Hh} the function values of the fine grid function are only copied.

Observing that the prolongator P_{hH} adds to the coarse grid Ω_H with $\frac{N}{4}$ inner points $\frac{3N}{4}$ new points, the number of operations for the linear interpolation P_{hH} defined in [6] (*9-point prolongation* in [9]) is : 2N.

When the residual $\phi^{(j-1)} - (I - b_0\tau^\nu J)x$ is determined by means of the

matrix $I - b_0\tau^{\nu}J$, the number of operations is : 10N. However, when the residual is determined by means of the matrix R, the number of operations is:

using ILU - 5 or ILU - 7 relaxation in TLA: 4N,

using ILU - 9 relaxation in TLA : 8N.

In table 4.2 the number of operations for the different parts of TLA in $I^pC_\rho I^s$ mode are listed, when ILU - 5, ILU - 7 and ILU -9 relaxation has been used (see also table 3.1). It is assumed that the residual is determined by means of the residual matrix R.

Table 4.2. Specification of the number of operations for each different part of TLA in $I^pC_\rho I^s$ mode with ILU - 5, ILU -7 and ILU -9 .

Part of TLA	TLA with ILU - 5	TLA with ILU - 7	TLA with ILU - 9
n_a	13N	17 N	25 N
n_b	6 .75 N	6 .75 N	10.75 N
n_c	2 .25 N	3 .25 N	4.25 N
n_d	3 .25 N	4 .25 N	6.25 N
n_e	3 N	3 N	3 N

From table 4.2 it follows that the computational work to perform one TLA iteration, i.e. W_{TLA} , with ILU - 5, ILU - 7 and ILU - 9 relaxation in TLA is $[p + s + (\frac{\rho-1}{4})] * 13\ N + 12N$, $[p + s + (\frac{\rho-1}{4})] * 17N + 13N$ and $[p + s + (\frac{\rho-1}{4})] * 25N + 18N$, respectively.

For example the number of operations W_{TLA} of one two-level iteration TLA in IC_4I mode with ILU - 7 relaxation, which is frequently used in our numerical experiments [6], is : 59.75N.

4.2. The decomposition on Ω_h and Ω_H

Let us denote the number of operations to perform the incomplete LU-decomposition on Ω_h and Ω_H by $(L^*U^*)_h + (L^*U^*)_H$. The number of operations to perform each particular decomposition on Ω_h and Ω_H is given in table 4.3.

Table 4.3. The number of operations to perform the ILU-decomposition on Ω_h and Ω_H for ILU - 5, ILU - 7 and ILU - 9.

Particular ILU-decomposition	$(L^*U^*)_h + (L^*U^*)_H$
ILU - 5	10 N
ILU - 7	21.25 N
ILU - 9	35 N

4.3. The evaluation of the function $\phi^{(j-1)}$

Suppose one f-evaluation f (the right-hand side of (1.1)) is equal to λ N operations on $\Omega_h(\lambda>0)$ and one Σ_n-evaluation (the right-hand side of (1.2)) is equal to δ N operations on $\Omega_h(\delta>0)$. Then the number of operations to perform the evaluation of $\phi^{(j-1)}$ (denoted by E, see section 2) occurring in (2.1) is : $(\lambda+\delta+13)N$.

4.4. The computational work per integration step

Let the evaluation of the matrices $I - b_0\tau^\nu J$ and $I - b_0\tau^\nu J_H$ be denoted by Jac_{Hh}. Then for a nonlinear problem the number of operations per integration step of the PCGC method, denoted by W_{PCGC}, can be given by the following expression

$$W_{PCGC} = Jac_{Hh} + (L^*U^*)_h + (L^*U^*)_H + m * E + r * W_{TLA},$$

where W_{TLA}, $(L^*U^*)_h + (L^*U^*)_H$ and E are defined in the sections 4.1, 4.2 and 4.3, respectively and the parameters r and m are both defined in section 2. In [6] the PCGC method has been compared with some other integration methods on the basis of computational units. For two parabolic PDE's in appendix B the number of operations per integration step is specified for these methods and the PCGC method.

5. NUMERICAL EXPERIMENTS

5.1. The test examples

All initial-boundary value problems chosen for our numerical experiments are defined on $0 \le t \le 1$ and

$$\Omega = \{(x_1,x_2) \mid \ 0 \le x_1,x_2 \le 1\},$$

and semi-discretized on a uniform grid Ω_h with mesh width h by standard symmetric differences and the coarse grid Ω_H has grid parameter $H = 2h$.

Our first example is a linear parabolic PDE (cf.[6]):

$$(5.1) \qquad \begin{cases} U_t = \alpha(U_{x_1x_1} + U_{x_2x_2}) - e^{-t}\,(4\alpha+x_1^2+x_2^2)\ , \ \alpha = 1,100, \\ U(x_1,x_2,t) = \alpha\, e^{-t}(x_1^2+x_2^2) + 1\ . \end{cases}$$

The second example is a nonlinear parabolic PDE (cf.[6]):

$$(5.2) \qquad \begin{cases} U_t = \left(\dfrac{\partial^2}{\partial x_1^2} + \dfrac{\partial^2}{\partial x_2^2}\right) U^5\ , \\ U(x_1,x_2,t) = \left[\dfrac{4}{5}(2t+x_1+x_2)\right]^{1/4}\ . \end{cases}$$

The initial and boundary conditions can be prescribed by providing the exact solution.

5.2. The numerical scheme

For parabolic PDE's, i.e. $\nu = 1$ in (1.1), we integrate the initial value problem (1.1) with the fourth order backward differentiation formula [11] which results in

$$(5.3) \qquad b_0 = \frac{12}{25}\ , \ \Sigma_n = \frac{1}{25}\,[48\,y_n - 36y_{n-1} + 16y_{n-2} - 3y_{n-3}]$$

in the iteration process (2.1).

In order to apply (5.3) four starting values are required which were obtained from the exact solution of the initial-boundary value problems. Furthermore, J and J_H were obtained by analytical differentiation. The half-bandwidth b defined in (3.2) is for J and J_H equal to $\frac{1}{h} - 1$ and $\frac{1}{2h} - 1$, respectively. In [6] the predictor formula starting the iteration process (2.1) is

$$y^{pred} = y_n. \tag{5.4}$$

In the sections (5.3), (5.4) and (5.5) we use only the predictor (5.4), whereas in section (5.6) we will consider also two other predictor formulas. For the PCGC method in $[E(I^p C_\rho I^s)^{r/m}]^m$ mode (see sections 2) the parameters p,ρ,s,r and m will be specified in the tables of results. In section 5.3 we use the ILU - 5, ILU - 7 and ILU - 9 relaxation in TLA. In the sections (5.4), (5.5) and (5.6) the ILU - 7 relaxation has only been used in TLA.

Furthermore, we use the notations:

$A(\tau)$ the number of correct digits at $t = 1$, i.e.

$$A(\tau) = -\,{}^{10}\log \| y_n - u(t_n) \|_\infty , \tag{5.5}$$

where $\| \ \|_\infty$ is the maximum norm and $u(t_n)$ denotes the exact solution of the PDE on the grid Ω_h at $t = t_n$.

r_{av} the average reduction factor of the two-level algorithm, i.e.

$$r_{av} = \left(\prod_{i=1}^{k-1} r_i \right)^{1/k-1} , \quad r_i = \frac{\| v^{(i+1)} - v^{(i)} \|_2}{\| v^{(i)} - v^{(i-1)} \|_2} , \quad i = 1,\ldots,k-1, \ k \geq 2, \tag{5.6}$$

where k is the number of TLA-iterations per Newton step (see section 2), $\| \ \|_2$ is the Euclidean norm, $v^{(i)}$ denotes the i-th iterand of TLA, $v^{(0)}$ is the starting value for the TLA-iteration and r_i is the reduction factor.

W_{TLA} the number of operations of one TLA-iteration.

$W_{10^{-1}}$ the number of operations for 10^{-1} reduction of the error by application of one TLA-iteration, i.e.

(5.7) $$W_{10}^{-1} = W_{TLA} / |^{10} \log r_{av}|.$$

In the experiments where the average reduction factor is considered the reduction factor r_i converges more or less to a limit value (see appendic C).

5.3. Numerical results of TLA with the three different ILU-iteration methods

In table 5.1 the results obtained by TLA in $I^p C_\rho I^s$ mode with ILU - 5, ILU - 7 and ILU - 9 relaxation are given for problem (5.1) with $\alpha = 100$ on a uniform grid Ω_h with $h = 1/32$. The number of inner points $N = (\frac{1}{h}-1)^2 = 961$.

Table 5.1. The average reduction factor r_{av}(5.6), the number of operations of one TLA-iteration W_{TLA} and W_{10}^{-1}(5.7) obtained by $I^p C_\rho I^s$ for problem (5.1) with $\alpha = 100$, $\tau = 1/4$, $h = 1/32$, $m = 1$ and $k = 8$.

TLA mode	r_{av}	W_{TLA}	W_{10}^{-1}
I^4C_8 with ILU - 5	0.288	86.75 N	160.47N
I^3C_6 with ILU - 7	0.108	85.25 N	88.20N
I^2C_4 with ILU - 9	0.116	86.75 N	92.73N
IC_4I with ILU - 5	0.489	47.75 N	153.69N
IC_4I with ILU - 7	0.216	59.75 N	89.78N
IC_4I with ILU - 9	0.115	86.75 N	92.36N
IC_5 with ILU - 5	0.45	38N	109.58N
IC_5 with ILU - 7	0.173	47N	61.68N
IC_5 with ILU - 9	0.085	68N	63.52N

In the TLA-algorithm in $I^p C_{2p}$ mode with $p = 2,3,4$ the number of operations of the ILU-5, ILU-7 and ILU-9 relaxation are approximately equal on the fine grid Ω_h and the coarse grid Ω_H. Table 5.1 illustrates that the I^3C_6 mode with ILU -7 relaxation is the most efficient one of the $I^p C_{2p}$

modes for this problem. When the same parameters p,ρ and s are used in TLA for the three different ILU-relaxation methods, then table 5.1 shows that the ILU-7 relaxation is again preferable. Notice that for the ILU-9 relaxation the preliminary work (decomposition) and the number of arrays to store L^*,U^*, R, L^*_H, U^*_H and $\tilde{R}_H$ are considerably more. In the following sections we will use only the ILU - 7 relaxation in the TLA algorithm.

5.4. The effect of the parameters p,ρ and s in the TLA algorithm

In table 5.2 the results are given obtained by the two-level algorithm in a particular mode with ILU - 7 relaxation for problem (5.1) with $\alpha = 100$ and $h = 1/32$. In order to compare the different TLA modes we have listed the number of operations of the ILU-7 relaxation on Ω_h and Ω_H. Note that the number of operations in the TLA modes is mainly determined by the ILU-7 relaxation on Ω_h and Ω_H (see table 4.2).

Table 5.2. Results for problem (5.1) with $\alpha = 100$, $\tau = 1/4$, $h = 1/32$ and $m = 1$.

M	PCGC method	r	A(1/4)	Nr. of operations of the ILU-7 relaxation on Ω_h and Ω_H
4	$E\,I^M$	0	-.89	68N
16			-.42	272N
32			.18	544N
2	$E(IC_4I)^r$	1	-.37	50N
4		2	.25	100N
8		4	1.51	200N
2	$E(IC_8I)^r$	1	.19	67N
4		2	1.32	134N
6		3	2.48	201N
8		4	3.68	268N
1	$E(IC_8)^r$	1	-.13	50N
2		2	1.23	100N
4		4	3.48	200N
1	$E(C_8I)^r$	1	-.26	50N
2		2	1.18	100N
4		4	3.39	200N

Additional experiments have shown that the accuracy gradually increases if the number of coarse grid corrections r increases until the limit value ($A(1/4) \simeq 4.7$) is reached. Increasing the number of operations to solve the coarse grid problem (2.3) (i.e. increasing ρ) improves the accuracy and the computational efficiency. The ILU-7 relaxation before rather than before and after the coarse grid correction is preferable. The results also show that for the accuracy it is better to iterate (with ILU-7 relaxation) before than after the coarse grid correction (cf.[4,10]).

The numerical results listed in table 5.3 show the dependence of the average reduction factor r_{av} on the number of iterations ρ on Ω_H in the evaluation of the coarse grid correction.

Table 5.3. The average reduction factor r_{av}(5.6), the number of operations of one TLA-iteration W_{TLA} and W_{10}-1(5.7) obtained by $I^p C_\rho I^s$ for problem (5.1) with $\alpha = 100$, $\tau = 1/4$, $k = 8$, $m = 1$ and ILU-7 relaxation.

h = 1/20				h = 1/32			
TLA-mode	r_{av}	W_{TLA}	W_{10}-1	TLA-mode	r_{av}	W_{TLA}	W_{10}-1
IC_1I	0.284	47N	85.97N	IC_4I	0.216	59.75 N	89.78N
IC_2I	0.14	51.25 N	60.02N	IC_5I	0.159	64N	80.14N
IC_3I	0.072	55.50 N	48.57N	IC_8I	0.064	76.75 N	64.29N
IC_4I	0.04	59.75 N	42.74N	$IC_{12}I$	0.022	93.75 N	56.56N
IC_5I	0.025	64N	39.95N	$IC_{13}I$	0.018	98N	56.17N
IC_6I	0.018	68.25 N	39.12N	IC_5	0.173	47N	61.68N
IC_8I	0.015	76.75 N	42.08N	IC_7	0.095	55.50 N	54.29N
$IC_{10}I$	0.015	85.25 N	46.74N	IC_8	0.072	59.75 N	52.29N

By increasing ρ for h = 1/20, 1/32 the average reduction factor r_{av} and W_{10}-1 of $I\,C_\rho\,I$ decrease until their minimal values are reached. Table 5.3 illustrates also that it is better to iterate before than before and after the

coarse grid correction.

In table 5.4 numerical results for the non-linear problem (5.2) obtained by $[EIC_4I]^m$ and $[EIC_8I]^m$ are listed. In order to compare both methods the $A(\tau)$ - values, the total number of evaluations of $\phi^{(j-1)}$ (ΣE, see section 4.3) and the total number of operations of the TLA iterations ΣW_{TLA} ($\Sigma W_{TLA} = (\frac{1}{\tau} - 3) * m * W_{TLA}$, see section 4.1) are listed.

Table 5.4. Results for problem (5.2) with $h = 1/32$, $\tau = 1/10$ obtained by $[EIC_4I]^m$ and $[EIC_8I]^m$.

$[EIC_4I]^m$				$[EIC_8I]^m$			
m	$A(\tau)$	ΣE	ΣW_{TLA}	m	$A(\tau)$	ΣE	ΣW_{TLA}
1	2.07	7	418.25N	1	2.89	7	537.25N
2	2.94	14	836.5N	2	3.8	14	1074.5N
3	3.75	21	1254.75N	3	4.78	21	1611.75N
4	4.56	28	1673N	4	5.76	28	2149N

From the results listed in table 5.4 it follows that increasing the number of operations to solve the coarse grid problem (2.3) improves the accuracy and the computational efficiency for this problem.

5.5. The effect of grid refinement

In table 5.5 we illustrate the dependence of r_{av} and W_{10}^{-1} of a two-level algorithm in $IC_\rho I$ mode on the grid parameter h. The average reduction factor r_{av} and W_{10}^{-1} increase when finer grids are used.

Table 5.5. Results for problem (5.1) with $\alpha = 100$, $\tau = 1/4$, $m = 1$, $k = 8$ and ILU-7 relaxation.

h	IC_4I		IC_8I	
	r_{av}	W_{10}^{-1}	r_{av}	W_{10}^{-1}
1/20	0.04	42.74N	0.015	42.08N
1/24	0.084	55.54N	0.018	43.99N
1/32	0.216	89.78N	0.064	64.29N
1/40	0.347	129.98N	0.155	94.79N
1/48	0.448	171.34N	0.258	130.44N

If ρ is sufficiently large chosen an upper-bound for the reduction factor of TLA can be found which is strictly less than 1 for all mesh sizes $h = H/2$ of the fine grid (see appendix C).

In table 5.6 the $A(\tau)$ - values for problem (5.1) with $\alpha = 100$ and $\tau = 1/4$ are illustrated obtained by $E(IC_4I)^4$ and $E(IC_8I)^4$ for a range of h-values. Both methods lose accuracy when the grid parameter h decreases.

Table 5.6. $A(\tau)$ - values for problem (5.1) with $\alpha = 100$, $\tau = 1/4$ obtained by $E(IC_4I)^4$ and $E(IC_8I)^4$ with ILU-7 relaxation.

h	$E(IC_4I)^4$	$E(IC_8I)^4$
1/20	4.83	4.70
1/24	3.19	4.71
1/32	1.51	3.68
1/40	.65	2.09
1/48	.16	1.19

5.6. Comparison of three different predictor formulas

The predictor formula (5.4) has been compared with two other predictor formulas. The first one is the *third order extrapolation formula*

$$(5.4)' \qquad y^{(pred)} = 4y_n - 6y_{n-1} + 4y_{n-2} - y_{n-3}.$$

The second formula is the *third order Adams - Bashforth method*

$$(5.4)'' \qquad y^{(pred)} = y_n + \frac{\tau}{12}\,[23\, f(t_n,y_n) - 48\, f(t_{n-1},y_{n-1}) + 5\, f(t_{n-2},y_{n-2})].$$

For the *method of successive corrections* the starting value $y^{(pred)}$ should be asymptotically stable (cf.[5]). Therefore, the formulas (5.4) and (5.4)' seem to be plausible choices and explicit linear multistep formulas are excluded as predictor formulas. The explicit formula (5.4)" introduces instabilities for large values of the spectral radius of the Jacobian matrix $\partial f/\partial y$.

In table 5.7 the effect of the predictor formulas (5.4),(5.4)' and (5.4)" is illustrated for the non-linear problem (5.2) in the PCGC method. Instability is indicated by an asterisk.

Table 5.7. Values of $A(\tau)$ obtained for problem (5.2) with $h = 1/20$ by $(EIC_4I)^m$ with ILU - 7 relaxation and the predictor formulas (5.4), (5.4)' and (5.4)".

τ	m	$y^{(pred)}$:(5.4)	$y^{(pred)}$: (5.4)'	$y^{(pred)}$:(5.4)"
	1	2.36	2.32	0.11
	2	2.71	4.17	3.21
1/5	3	3.46	5.01	4.58
	4	4.06	5.02	5.02
	5	4.64	5.02	5.02
	1	2.91	5.65	*
	2	3.82	6.54	*
1/10	3	4.82	6.67	6.67
	4	5.82	6.67	6.67
	5	6.67	6.67	6.67

The results listed in table 5.7 indicate that the third order extrapolation formula (5.4)' seems to be a more suitable choice than the predictor formula (5.4). Note that the number of operations per integration step to calculate the predictor (5.4)' is 6 N on a uniform grid Ω_h with N inner points. The third order Adams-Bashforth formula (5.4)" should introduce instabilities which is confirmed by the results in table 5.7.

6. CONCLUDING REMARKS

For the problems under consideration it is better to choose ILU-7 relaxation in the two-level algorithm than ILU-5 or ILU-9 relaxation. However, the best choice of the parameters and operators in a two-level method and a multi-grid method depends on the problem under consideration (cf.[9]).

The convergence rate of the two-level algorithm in a particular mode depends on the number of iterations ρ on the coarse grid and the grid parameter h. Therefore, the PCGC method should be implemented as a multi-grid method as described in [2,9]. Increasing the number of iterations ρ on the coarse grid improves the accuracy and the computational efficiency until the coarse grid problem (2.3) is solved with negligible error. The number of iterations ρ to solve (2.3) with negligible error depends on the grid parameter h.

For the accuracy it is better to iterate before than after the coarse grid correction.

For the model problem (5.1) the discrete Fourier analysis gives quite satisfactory results for small h (see appendix A and C).

Using the third order extrapolation formula (5.4)' as predictor formula in the PCGC method instead of the zero order extrapolation formula (5.4) is worthwhile for the problems here considered.

The PCGC method requires the storage of $I - b_0 \tau J$, $I - b_0 \tau J_H$, L^*, U^*, R, L^*_H, U^*_H, $\tilde{R}_H$, $\phi^{(j-1)}$ and $y_{n+1-\ell}$ for $\ell = 0(1)\ 4$, when the fourth order backward differentiation formula (5.3) is chosen as the implicit formula (1.2). Then to implement the PCGC method with ILU-7 relaxation we need $23\frac{1}{2}$ arrays of length N, where N is the number of interior grid points on Ω_h.

ACKNOWLEDGEMENT

The author would like to thank Prof.Dr. P.J. van der Houwen for his constructive remarks and valuable advice.

REFERENCES

[1] BRANDT, A., *Multi-level adaptive solutions to boundary value problems*, Math. Comp. 31 (1977), pp.333-390.

[2] HEMKER, P.W., *Introduction to multi-grid methods*, Nieuw Arch. Wiskunde (3), XXIX (1981), pp. 71-101.

[3] HEMKER, P.W., *The incomplete LU-decomposition as relaxation method in multi-grid algorithms*, In : *Boundary and Interior Layers- Computational and Asymptotic methods*, ed. J.J.H. Miller, Boole Press, Dublin (1980), pp. 306-311.

[4] HEMKER, P.W., *Lecture notes on a seminar on multiple grid methods*, Report NN 24/81, Mathematisch Centrum, Amsterdam (1981).

[5] HOUWEN, P.J. VAN DER, *Multistep splitting methods of high order for initial value problems*, SIAM J.Numer. Anal., 17(1980), pp. 410-427.

[6] HOUWEN, P.J. VAN DER & H.B. DE VRIES, *Preconditioning and coarse grid corrections in the solution of the initial value problems for non-linear partial differential equations*, Report NW 95/80, Mathematisch Centrum, Amsterdam (1980), (to appear in SIAM J. on Sci. and Stat. Comp., (1982)).

[7] MEIJERINK, J.A. & H.A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. 31 (1977), pp. 148-162.

[8] MOL, W.J.A., *Numerical solution of the Navier-Stokes equations by means of a multigrid method and Newton iteration*, Report NW 92/80, Mathematisch Centrum, Amsterdam (1980).

[9] MOL, W.J.A., *On the choice of suitable operators and parameters in multigrid methods*, Report NW 107/81, Mathematisch Centrum, Amsterdam (1981).

[10] MOL, W.J.A., *Smoothing and coarse grid approximation properties of multigrid methods*, Report NW110/81, Mathematisch Centrum, Amsterdam (1981).

[11] LAMBERT, J.D., *Computational methods in ordinary differential equations*, John Wiley & Sons, London, 1973.

[12] SOMMEIJER, B.P. & P.J. VAN DER HOUWEN, *On the economization of stabilized Runge-Kutta methods with applications to parabolic initial value problems*, Z. Angew. Math. Mech., 61(1981), pp. 105-114.

[13] SOMMEIJER, B.P. & J.G. VERWER, *A performance evaluation of a class of Runge-Kutta-Chebyshev methods for solving semi-discrete parabolic differential equations*, Report NW 91/80, Mathematisch Centrum, Amsterdam (1980).

APPENDIX

A. THE CONVERGENCE RATE OF ILU-5, ILU-7 AND ILU-9 RELAXATION

For a range of h-values problem (5.1) with $\alpha = 100$ is discretized with respect to its space variables which results in systems of ODE's. In order to integrate these systems we choose again the fourth order backward differentiation formula (5.3) and we put $\tau = 1/4$. The linear equations are solved with ILU-5, ILU-7 and ILU-9 relaxation. The coefficient a_1 in (3.6) assumes the form $-12/h^2$.

For $-\pi < \theta_i = \omega_i\, h\pi < \pi$, $\omega_i \in \mathbb{Z} \setminus \{0\}$, $i = 1,2$ it can be numerically veryfied that for this range of h values the convergence factors $\mu_5(\theta), \mu_7(\theta)$ and $\mu_9(\theta)$ (see section 3.2) reach their maxima $\hat{\mu}_5, \hat{\mu}_7$ and $\hat{\mu}_9$, respectively, in $(\theta_1,\theta_2) = (\pi h, \pi h)$. The graphs of the functions $\hat{\mu}_5$, $\hat{\mu}_7$ and $\hat{\mu}_9$ are given in figure A.1.

Experimentally the reduction factor of the ILU-iteration, i.e. the spectralradius λ of $(L^*U^*)^{-1}R$, is calculated for problem (5.1) with $\alpha = 100$ and $\tau = 1/4$ by

$$\lambda = \frac{\|x_{i+1}-x_i\|_2}{\|x_i-x_{i-1}\|_2} \qquad ,i \gg 1, \tag{A.1}$$

where $\|.\|_2$ is the Euclidean norm and x_i is the i-th iterand of (2.7) for ILU-5, ILU-7 and ILU-9, respectively. These numerical values for the spectral radius of $(L^*U^*)^{-1}R$ in ILU-5, ILU-7 and ILU-9 are obtained after performing a sufficient number of iterations steps with (2.7), i.e. the ratio (A.1) has more or less converged to a limit value. In figure A.1 the reduction factors (A.1) for the ILU-5, ILU-7 and ILU-9 iteration are denoted by x, o and •, respectively.

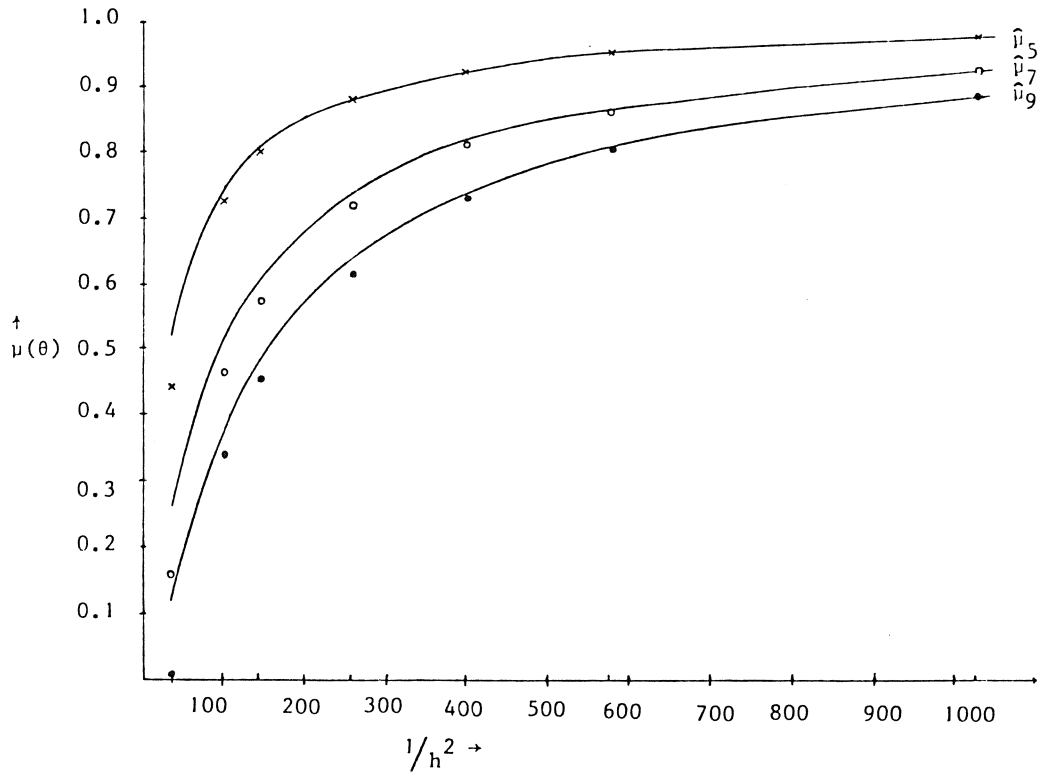

Figure A.1. The curves of the maximal convergence factors $\hat{\mu}_5$, $\hat{\mu}_7$ and $\hat{\mu}_9$ of ILU-5, ILU-7 and ILU-9 relaxation, respectively. The reduction factors (A.1) of ILU-5, ILU-7 and ILU-9 relaxation are denoted by x, o and •, respectively.

The Fourier analysis gives quite satisfactory results for $h \leq 1/16$. For $h > 1/16$ figure A.1 shows that for increasing values of h the convergence rate predicted by the theory differs still more from the experimental values, which is due to the boundary conditions. It should be noted that for $a_1 = -\frac{12}{h^2} \ll -1$ the same conclusions can be drawn by considering the Poisson equation [8,10], i.e. $a_0 = 4, a_1 = -1$.

B. COMPARISON OF THE EFFICIENCY OF THE PCGC METHOD, THE RKC METHOD AND THE PR METHOD

In [6] the PCGC method with the predictor (5.4) in $(E(IC_4I)^{r/m})^m$ mode has been compared with the second order one-step Runge-Kutta-Chebyshev method (RKC method [12]) and the Preconditioned Richardson method (PR method). In [13] the numerical results with the RKC method are given for the test examples (5.1) with $\alpha = 1$ and (5.2). The computational effort is measured by computational units, which are defined differently for each method. Here we give a more detailed discussion of these computational units, i.e. we compare the 3 methods on the basis of arithmetic operations (see also section 4).

The PR method is also based on the same Newton iteration (2.1). However, for the PR method the inner iteration is defined differently (see [6]) and the ILU-decomposition and the evaluation of the Jacobian matrix are only performed on the fine grid Ω_h. The PR method will be denoted by $(E\,I^q)^m$ where q is the number of iteration steps to solve each linear system to be specified for each problem.

Although in the numerical experiments with the PR and the PCGC method the number of integration steps is $(\tau^{-1}-3)$, we assume that in the comparison of the three methods the number of integration steps in the PR and the PCGC method is τ^{-1}. The tables in [6], which illustrated the $A(\tau)$-values and the corresponding computational work of the three methods, are based on the above assumption.

We suppose that one f-evaluation (denoted by f) is equal to λ N operations on Ω_h (see section 4.3). From (5.3) it follows that one Σ_n-evaluation (see section 4.3) is equal to 7 N operations on Ω_h.

B.1. The RKC method

The number of operations in (almost)each stage of the RKC-formula [12] is : $(\lambda+9)$ N.

B.2. The PR method

B.2.1. The decomposition

In the incomplete LU-decomposition based on P_0(3.2) of $I - b_0\tau J$ on Ω_h the residual matrix R is not computed. Therefore it is convenient to write the incomplete decomposition in the form LDU, where D is a diagonal matrix (see MEIJERINK & VAN DER VORST [7]). In this case we need only compute and store the matrix D. On Ω_h the number of operations of this decomposition (denoted by $(L^*U^*)_h$) is : 6 N.

B.2.2. The evaluation of the function $\phi^{(j-1)}$

The number of operations to perform the evaluation of $\phi^{(j-1)}$ (denoted by E) occurring in (2.1) is (cf.section 4.3) : $(\lambda+20)$ N.

B.2.3. The iteration step

The number of operations on Ω_h for one iteration step I is : 25 N.

B.3. Specification of the computational units

For the linear problem (5.1) with $\alpha = 1$ and the non-linear problem (5.2) we assume that one f-evaluation f is equal to 12 N and 11 N operations on Ω_h, respectively. In the PR method and the PCGC method the computational work for the evaluation of the Jacobian matrices and the incomplete decompositions are not taken into account for the *linear problem* (5.1), because for this problem these calculations are required only one. In the RKC method the evaluation of the spectral radius of the Jacobian matrix is neglected and in the PR method all initial work for estimating the iteration parameters is not taken into account.

For the *non-linear problem* we assume that for the evaluation of the Jacobian matrices in the PR and the PCGC method the number of operations per integration step is 7 N and 8.75 N, respectively.

In order to define the computational units we introduce the following notations (cf.[6]):

matvec a matrix-vector multiplication on Ω_h

sol the solution of $L^*U^*\ y = b$ on Ω_h.

In table B.1 the number of operations per stage in the RKC method, the number of operations per integration step in the PR and the PCGC method and the computational units defined in [6] are listed for both problems.

Table B.1. The number of operations per stage in the RKC method and per integration step in the PR and the PCGC method for problem (5.1) with $\alpha = 1$ and (5.2). The computational units are defined differently for each method in [6].

Linear problem (5.1) with $\alpha = 1$ and m = 1		
Method	Nr. of operations	Computational unit
RKC	21 N	4f
$E\ I^q$ (q=14)	382 N	$\frac{1}{5}$ [f+matvec+q(matvec+sol)]
$E(IC_4I)^r$ (r=3,4)	32N+r*59.75N	$\frac{1}{r}$ [f+matvec]+C_4+3matvec+2sol
Nonlinear problem	(5.2) with m = 1,2,3, 4.	
Method	Nr. of operations	Computational unit
RKC	20N	10f
$(EI^q)^m$ (q=9)	13N + m*256N	$\frac{1}{2}$ [f+(q+1)matvec+q sol + $\frac{1}{m}$ * $(L^*U^*)_h$]
$(EIC_4I)^m$	30N + m* 90.75N	f+4matvec+2sol+C_4+ $\frac{1}{m}$ * $((L^*U^*)_h+(L^*U^*)_H)$

The computational units for problem (5.1) are based on the assumption that the number of operations in 4-stages with RKC, in 1/r integration step with $E(IC_4I)^r$ for r = 3,4 and in 1/5 integration step with $E\ I^{14}$ is approximately equal. In the definition of the units for problem (5.2) we assume that the number of operations in 10 stages with RKC, in 1/2m integration step with $[E\ I^9]^m$ and in 1/m integration step with $(EIC_4I)^m$ for m = 1,2,3,4

is approximately equal.

C. THE REDUCTION FACTOR OF TLA

In order to demonstrate that the reduction factor r_i (cf.(5.6)) of TLA converges to a limit value r_i^* we choose the linear problem (5.1) with $\alpha = 100$. In table C.1 we give the reduction factor r_i, the average reduction factor r_{av} and the errors $\|v^{(i)} - v^{(i-1)}\|_2$, where $v^{(i)}$ denotes the i-th iterand of TLA in IC_8I mode. The numbers in the parentheses stand for exponents of 10.

Table C.1. The reduction factor r_i, the average reduction factor r_{av} and the errors $\|v^{(i)} - v^{(i-1)}\|_2$ (i=1,2,...,8) for problem (5.1) with $\alpha = 100$, $\tau = 1/4$ obtained by $E(IC_8I)^i$ with ILU-7 relaxation.

h = 1/32			h = 1/40			h = 1/48		
i	$\|v^{(i)}-v^{(i-1)}\|_2$	r_i	i	$\|v^{(i)}-v^{(i-1)}\|_2$	r_i	i	$\|v^{(i)}-v^{(i-1)}\|_2$	r_i
1	2.41(+2)	0.044	1	2.93(+2)	0.097	1	3.37(+2)	0.148
2	1.07(+1)	0.067	2	2.85(+1)	0.167	2	5.0 (+1)	0.28
3	7.2 (-1)	0.069	3	4.75	0.168	3	1.4 (+1)	0.283
4	4.95(-2)	0.069	4	7.98(-1)	0.168	4	3.95	0.283
5	3.41(-3)	0.069	5	1.34(-1)	0.168	5	1.12	0.283
6	2.35(-4)	0.069	6	2.25(-2)	0.168	6	3.17(-1)	0.283
7	1.62(-5)	0.069	7	3.77(-3)	0.168	7	8.98(-2)	0.283
8	1.11(-6)		8	6.33(-4)		8	2.54(-2)	
$r_{av} = 0.064$			$r_{av} = 0.155$			$r_{av} = 0.258$		

The reduction factor of the two-level algorithm IC_8I converges for h = 1/32, 1/40 and 1/48 to the limit values 0.069, 0.168 and 0.283, respectively. When the number of iterations ρ on Ω_H are not sufficient to solve the coarse grid problem (2.3) with negligible error, then for small h the convergence rate of TLA in $I^pC_\rho I^s$ mode is approximately equal to $[\mu_7(\pi h,\pi h)]^{p+s}\ [\mu_7(2\pi h,2\pi h)]^\rho$ (see section 2 and table 3.3), i.e. the

convergence rate of TLA will be determined by the low frequency components in the error.

For example, using table 3.3 the estimated convergence rate of IC_8I is 0.077, 0.179 and 0.294 for h = 1/32, 1/40 and 1/48, respectively. Comparing these values with the limit values r_i^* for h = 1/32, 1/40 and 1/48, we observe that we have overestimated the convergence rate with 11.6%, 6.7% and 3.9%, respectively. In a similar way the estimated convergence rate of IC_4I is 0.255, 0.401 and 0.522 for h = 1/32, 1/40 and 1/48, respectively, whereas experimentally the convergence rate of IC_4I tends to 0.235, 0.3863 and 0.512 for h = 1/32, 1/40 and 1/48, respectively. Thus in this case the convergence rate is overestimated with 8.5%, 3.8% and 2%, respectively.

In section 3.2 we have derived the smoothing factor $\bar{\mu}_7$ of the ILU-7 iteration for a model problem. For the model problem (5.1) with $\alpha = 100$ we illustrate that the reduction factor r_i (5.6) of the TLA algorithm in $I^pC_\rho I^s$ mode converges more or less to the limit value $(\bar{\mu}_7)^{p+s}$ with $\bar{\mu}_7$ listed in table 3.2, when the coarse grid problem (2.3) is solved with negligible error after ρ iterations with (2.8).

In table C.2a and C.2b for problem (5.1) with $\alpha = 100$ the reduction factor r_i and the errors $\|v^{(i)} - v^{(i-1)}\|_2$ are listed for $E(IC_{16}I)^i$ and $E(IC_{16})^i$, respectively.

Table C.2a. The reduction factor r_i and the error $\|v^{(i)} - v^{(i-1)}\|_2$ for problem (5.1) with $\alpha = 100$, $\tau = 1/4$ obtained by $E(IC_{16}I)^i$ with ILU-7 relaxation.

i	h = 1/20		h = 1/32	
	$\|v^{(i)} - v^{(i-1)}\|_2$	r_i	$\|v^{(i)} - v^{(i-1)}\|_2$	r_i
1	1.5 (+2)	0.0118	2.48 (+2)	0.0112
2	1.76	0.0143	2.76	0.0132
3	2.51(-2)	0.015	3.65(-2)	0.0141
4	3.76(-4)	0.0155	5.13(-4)	0.0149
5	5.85(-6)	0.0161	7.63(-6)	0.0157
6	9.39(-8)	0.0165	1.19(-7)	0.0164
7	1.55(-9)		1.96(-9)	

The results in table C.2a and table C.2b show that the reduction factor r_i of $IC_{16}I$ and IC_{16} converges more or less to $(\bar{\mu}_7)^2 \cong 0.01585$ and $\bar{\mu}_7 \cong 0.1259$, respectively. It should be noted that the smoothing analysis (see section 3.2) applies to Toeplitz-matrices without considering the boundaries. From the results listed in table C.2a and C.2b the average reduction factor r_{av} (5.6) of $IC_{16}I$ and IC_{16} can be derived. The average reduction factor r_{av} of $IC_{16}I$ is 0.0148 and 0.0141 for h = 1/20 and 1/32, respectively and r_{av} of IC_{16} is 0.0792 and 0.0799 for h = 1/20 and 1/32, respectively. Notice that r_{av} of $IC_{16}I$ and r_{av} of IC_{16} is smaller than $(\bar{\mu}_7)^2$ and $\bar{\mu}_7$, respectively.

Table C.2b. The reduction factor r_i and the error $\|v^{(i)} - v^{(i-1)}\|_2$ for problem (5.1) with $\alpha = 100$, $\tau = 1/4$ obtained by $E(IC_{16})^i$ with ILU-7 relaxation.

i	h = 1/20		h = 1/32	
	$\|v^{(i)} - v^{(i-1)}\|_2$	r_i	$\|v^{(i)} - v^{(i-1)}\|_2$	r_i
1	1.48 (+2)	0.0276	2.46 (+2)	0.0229
2	4.09 (+1)	0.0412	5.64	0.0422
3	1.68 (-1)	0.0654	2.38 (-1)	0.0652
4	1.1 (-2)	0.0751	1.55 (-2)	0.0767
5	8.26 (-4)	0.0909	1.19 (-3)	0.0927
6	7.51 (-5)	0.1	1.1 (-4)	0.1032
7	7.51 (-6)	0.105	1.14 (-5)	0.1089
8	7.89 (-7)	0.1075	1.24 (-6)	0.1117
9	8.48 (-8)	0.1091	1.39 (-7)	0.1133
10	9.25 (-9)	0.1102	1.57 (-8)	0.1144
11	1.02 (-9)	0.112	1.8 (-9)	0.1164
12	1.14 (-10)		2.09(-10)	