stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 148/83 FEBRUARI

H.B. DE VRIES

THE MULTI-GRID METHOD IN THE SOLUTION OF TIME-DEPENDENT NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, Kruislaan 413, Amsterdam, The Netherlands.

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

1980 Mathematics subject classification: 65F10, 65M20

The multi-grid method in the solution of time-dependent nonlinear partial differential equations *)

by

H.B. de Vries

ABSTRACT

The numerical solution is discussed of nonlinear, time-dependent partial differential equations. By applying the method of lines such a partial differential equation is converted into a system of ordinary differential equations to which an implicit linear multistep method is applied. 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 multi-level algorithm. Numerical examples are given and a comparison is made with other integration techniques.

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

---

*) This report will be submitted for publication elsewhere.

1. INTRODUCTION

Let the system of ordinary differential equations (ODE's)

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

with prescribed values for y^k (and dy^k/dt) at $t = t_0$ originate from the semi-discretization on a uniform grid Ω^k (with grid parameter h^k) of a parabolic (ν=1) or hyperbolic (ν=2) two-dimensional partial differential equation (PDE). Here, the index k refers to the grid Ω^k, i.e. k denotes the level of discretization. By applying a linear multistep method to this equation we are asked to solve at each time step the system of equations

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

where y^k_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 (1.2) is identified with y^k_{n+1}.

Using *Newton iteration* the nonlinear implicit relations (1.2) are replaced by a sequence of linear equations. In section 2 we describe a *multi-grid method* for the solution of these linear equations.

The computational work involved in the Newton iteration and the multi-grid method is considered in section 3.

Finally, in section 4 numerical experiments are reported for a number of semi-discrete parabolic equations (ν=1). Also comparisons are given with the PCGC method [5] and the SC method [6].

2. THE MULTI-GRID METHOD

Using the *modified Newton-Raphson* process we replace the system of equations (1.2) by a sequence of m systems of linear equations on Ω^k[5,9]:

$$(y^k)^{(0)} = (y^k)^{(\mathrm{pred})},$$

$$[I^k - b_0\tau^{\nu}J^k](y^k)^{(j)} = (\Phi^k)^{(j-1)},\ j = 1,\ldots,m, \tag{2.1}$$

$$J^k = \frac{\partial f^k}{\partial y^k}\,(t_{n+1}, (y^k)^{(0)}),$$

$$(\Phi^k)^{(j-1)} = \Sigma_n^k + b_0\tau^\nu[f^k(t_{n+1}, (y^k)^{(j-1)}) - J^k(y^k)^{(j-1)}],$$

where $(y^k)^{(pred)}$ is obtained by some predictor formula on Ω^k, J^k is the Jacobian matrix on Ω^k and Σ_n^k denotes the right-hand side of (1.2) on Ω^k.

In the outer iteration (2.1) each of the systems of linear equations is solved by the iterative use of a *multi-grid method* [2] (inner iteration). The multi-grid method uses a coarse to fine sequence of computational grids, viz. $\Omega^k, \Omega^{k-1}, \ldots, \Omega^0$ (Ω^k is the finest grid and Ω^0 is the coarsest grid), whereas in the two-grid method the grids Ω^k and Ω^{k-1} are only used [2,8]. For the computational grids Ω^ℓ (ℓ=0,1,...,k) the grid parameter h^ℓ is defined by $h^\ell = h^{\ell-1}/2$ for ℓ = k, k-1,...,1. In the two-level algorithm (cf.[5,9]) we have to solve a discretized problem on the coarse grid Ω^{k-1}. The multi-level algorithm approximates the solution of this problem by application of a number of iteration steps of the same algorithm on the coarse level, i.e. the multi-level algorithm is the recursive application of the two-level algorithm. Now we only have to solve directly a discretized problem on the very coarsest grid Ω^0.

Before we describe the multi-level algorithm we introduce the *restriction operator* R^ℓ and *prolongation operator* P^ℓ for ℓ = 1,...,k (cf.[5]):

$$R^\ell: V^\ell \to V^{\ell-1}, \tag{2.2a}$$

$$P^\ell: V^{\ell-1} \to V^\ell, \tag{2.2b}$$

where V^j is a grid function defined on level j for j = 0(1)k.

In the multi-grid method we consider the sequence of equations

$$[I^\ell - b_0\tau^\nu J^\ell]\,x^\ell = \phi^\ell,\quad J^\ell = \frac{\partial f^\ell}{\partial y^\ell}\,(t_{n+1}, (y^\ell)^{(0)}),\quad \ell = 0(1)k,$$
$$(y^\ell)^{(0)} = I^{k\ell}(y^k)^{(0)} \text{ for } \ell < k, \tag{2.3}$$

where for $\ell < k$ $I^{k\ell}$ represents a simple transfer of values of the coarser grid Ω^{ℓ} from the corresponding points in the finest grid Ω^{k} (called in [1] injection). The functions ϕ^{ℓ} are defined by

$$\phi^{k} = (\Phi^{k})^{(j-1)},$$
$$\phi^{\ell-1} = R^{\ell}(\phi^{\ell}-(I^{\ell}-b_0\tau^{\nu}J^{\ell})\tilde{x}^{\ell}) \text{ for } \ell = k,\ldots,1, \tag{2.4}$$

where $\tilde{x}^{\ell}$ is an approximation of the solution of (2.3) to be specified later.

For the solution of (2.3) we consider *iterative processes* based on an *incomplete* LU-*decomposition* (ILU) of $I^{\ell} - b_0\tau^{\nu}J^{\ell}$ on the levels $\ell = 1(1)k$:

$$(x^{\ell})^{(i)} = [\tilde{L}^{\ell}\tilde{U}^{\ell}]^{-1}\,[\tilde{R}^{\ell}(x^{\ell})^{(i-1)} + \phi^{\ell}], \tag{2.5}$$

where $\tilde{L}^{\ell}\tilde{U}^{\ell} - \tilde{R}^{\ell} = I^{\ell} - b_0\tau^{\nu}J^{\ell}$. The iteration method (2.5) (called in [2] relaxation method) can also be obtained by applying iterative refinement to the preconditioned system (2.3) (cf. [5]). In the experiments we use the ILU-7 relaxation method as described in [5,9]. Other possible relaxation methods are ILU-5 and ILU-9 (see [9]). These relaxation methods are suitable when the components in the right-hand side f^{k} of (1.1) are coupled according to a five-point molecule.

Furthermore, we assume that on the coarsest grid Ω^{0} the equation (2.3) with $\ell = 0$ is solved *exactly* by a (complete) LU-decomposition, i.e.

$$x^{0} = [L^{0}U^{0}]^{-1}\,\phi^{0},\quad L^{0}U^{0} = I^{0} - b_0\tau^{\nu}J^{0}. \tag{2.6}$$

The multi-level algorithm (MLA) can be described in quasi-Algol as:

```
proc MLA = (integer ℓ,p,q,s, vector y^ℓ,φ^ℓ) vector:
begin

vector    x^ℓ, φ^(ℓ-1), z^(ℓ-1);

if        ℓ = 0
```

then $x^0 := [L^0U^0]^{-1}\ \phi^0$

else $x^\ell := y^\ell$;

for i *to* p *do* $x^\ell := [\tilde{L}^\ell\tilde{U}^\ell]^{-1}\ [\tilde{R}^\ell x^\ell + \phi^\ell]$ *od*;

$\phi^{\ell-1} := R^\ell(\phi^\ell - [I^\ell - b_0\tau^\nu J^\ell]x^\ell)$;

$z^{\ell-1} := 0$;

for i *to* q *do* $z^{\ell-1} :=$ MLA$(\ell-1,p,q,s,z^{\ell-1},\phi^{\ell-1})$ *od*;

$x^\ell := x^\ell + P^\ell\ z^{\ell-1}$;

for i *to* s *do* $x^\ell := [\tilde{L}^\ell\tilde{U}^\ell]^{-1}\ [\tilde{R}^\ell x^\ell + \phi^\ell]$ *od*;

fi;

x^ℓ

end;

Notice that *proc* MLA is a recursive procedure.

One multi-level iteration, i.e. one execution of MLA, will be denoted by

(2.7) MLA (k,p,q,s),

where k indicates the highest level (finest grid Ω^k) and p,q and s are specified in *proc* MLA. The multi-level algorithm on level ℓ consists of:

i) p relaxation sweeps on level ℓ (pre-relaxation).

ii) a coarse grid correction (cf.[2,5]), consisting of:

  a) computation of $\phi^{\ell-1}$.

  b) approximation of the solution of (2.3) on level $(\ell-1)$, by either q sweeps of MLA on level $(\ell-1)$ or if $\ell = 1$ by (2.6).

  c) prolongation of the correction to level ℓ and addition of the correction to the latest approximate solution on level ℓ.

iii) q relaxation sweeps on level ℓ (post-relaxation).

The multi-level algorithm MLA is the linear variant of the multi-grid method and

has a fixed strategy, i.e. the iterations are controled by the fixed numbers p, q and s. Brandt [1] calls the linear variant the CS-algorithm. In the numerical experiments several choices for the parameters p, q and s will be considered. When p = 0, q = 1 and s ≥ 1 the multi-level algorithm MLA can be implemented in another way. Then the recursive structure of MLA can be avoided (cf.[8]).

REMARK 2.1. From (2.5) it follows that for ℓ = 1(1) k

$$\phi^{\ell} - [I^{\ell}-b_0\tau^{\nu}J^{\ell}](x^{\ell})^{(i)} = \tilde{R}^{\ell}[(x^{\ell})^{(i)} - (x^{\ell})^{(i-1)}],\ i \geq 1. \tag{2.8}$$

Since $\tilde{R}^{\ell}$ has usually less non-zero diagonals than the original matrix, this is a cheap way to compute the residual $\phi^{\ell} - [I^{\ell}-b_0\tau^{\nu}J^{\ell}]\ x^{\ell}$. In the numerical experiments the residual is computed by means of $\tilde{R}^{\ell}$. For instance, choosing p = 1 in *proc* MLA the residual is equal to $\tilde{R}^{\ell}(x^{\ell}-y^{\ell})$. For p = 0, i.e. only post-relaxation in MLA, we need an extra array for storage when the residual is computed by means of $\tilde{R}^{\ell}$.

In order to describe the PMG (*Preconditioning and Multi-Grid*) method we introduce the notations:

E evaluation of the function $(\Phi^k)^{(j-1)}$ defined in (2.1).

m number of right-hand side evaluations $(\Phi^k)^{(j-1)}$ per integration step, i.e. the number of Newton iterations (2.1) per integration step.

M number of MLA iterations on level k per Newton step.

A particular PMG method is now denoted by

$$[E(MLA(k,p,q,s))^M]^m. \tag{2.9}$$

The different variants of the multi-level algorithm are determined by the fixed numbers p, q and s and by selecting different procedures for the ILU-relaxation (2.5), the restriction (2.2a) and the prolongation (2.2b). We select the *weighted restriction* R^{ℓ} and *linear interpolation* P^{ℓ} as defined in [5] (called in [4,8] 9-point restriction and 9-point prolongation, respectively). Most work in MLA is spent with relaxation. From the theoretical and numerical results reported in [9] it follows that in two-level algorithm the ILU-7 relaxation is to be preferred to the ILU-5 and ILU-9

relaxation for the problems under consideration. In [10, Appendix A] we illustrate also by a few numerical experiments that ILU-7 relaxation appears to be the best choice in the multi-level algorithm. Therefore, we select the *ILU-7 relaxation.*

Let η be the solution of equation (1.2) and define the iteration error $\varepsilon_j = (y^k)^{(j)} - \eta$. Then it is easily verified that the final iteration error of (2.1) is $\varepsilon_m = O(\tau^{\hat{p}+\nu(m+1)})$ as $\tau \to 0$, where $\hat{p}$ is the order of accuracy of the predictor formula used in (2.1) (cf.[5]). Thus, the *order of consistency* of the PMG method is given by

$$(2.10) \quad p^* = \min (\tilde{p}, m\nu+\hat{p}),$$

where $\tilde{p}$ is the order of accuracy of the generating linear multistep method (1.2).

3. THE COMPUTATIONAL WORK OF THE PMG METHOD

In this section an estimate will be derived for the computational work of the PMG method. In this estimate the computation of matrix-indices (array subscripts) and other overhead costs are neglected (see also [4,8]). An operation will be defined as an element from the set $\{+,-,*,/\}$. Let the uniform grid Ω^k for two-dimensional PDE's have N inner points (in the experiments $N = ((1/h^k)-1)^2$), and the uniform grids Ω^ℓ with $h^\ell = 2^{k-\ell} h^k$ have approximately $N/4^{k-\ell}$ inner points for $\ell = 0(1)k-1$.

3.1. The computational work per integration step

In order to describe the computational work per integration step we introduce the following notations:

W_{PRED} the computational work to compute $(y^k)^{(pred)}$ (see (2.1))

W_{JACE} the evaluation of the matrices $I^\ell - b_0\tau^\nu J^\ell$ for $\ell = 0(1)k$

W_{IDEC} the computational work of the ILU-decomposition on Ω^ℓ for $\ell = 1(1)k$ (see section 2)

W_E the evaluation of $(\Phi^k)^{(j-1)}$ (see (2.1))

W_{MLA} the computational work to perform one MLA-iteration (or cycle) on level k.

For a nonlinear problem the computational work per integration step of the PMG method, denoted by W_{PMG}, can be given by the following expression

$$(3.1) \qquad W_{PMG} = W_{PRED} + W_{JACE} + W_{IDEC} + m*W_E + m*M*W_{MLA},$$

where m and M are defined in section 2. Notice that in case of linear problems the matrices $I^{\ell} - b_0\tau^{\nu}J^{\ell}$ and their decompositions are determined once. We assume that the coarsest grid Ω^0 is coarse enough (i.e., k is sufficiently large chosen) to make the solution of its algebraic system (2.3) with $\ell = 0$ inexpensive compared with one relaxation sweep over the finest grid Ω^k [1,8].

3.2. The computational work of one MLA-iteration

Here we derive an estimate of the computational work W_{MLA} to perform one MLA-iteration. It will be assumed that there are at least more than two grids, i.e. $k > 1$. The computational work in the multi-level algorithm is determined by (p+s) - relaxation sweeps on the levels $\ell = 1(1)k$, the computation of $\phi^{\ell-1}$ for $\ell = 1(1)k$, and the prolongation and addition of the correction on the levels $\ell = 1(1)k$.

The number of operations of the ILU-7 relaxation on Ω^k is equal to 17N (cf.[9]). Using the matrix $\tilde{R}^k$ to compute the residual $\phi^k - (I^k - b_0\tau^{\nu}J^k)\, x^k$ (see Remark 2.1) the number of operations is equal to 4N. Further, the numbers of operations for the restrictor R^k and prolongator P^k are 2.75N and 2N, respectively.

Then, the number of operations in one MLA-iteration is

$$(3.2) \qquad W_{MLA} = \frac{4}{4-q}\,[9.75+(p+s)17]\,N, \quad q < 4.$$

REMARK 3.1. Note that in the special case p = 0 and q = 1 the multi-level algorithm can be implemented without using the recursive structure (see also section 2). Then writing *proc* MLA in this simple way the number of operations in one MLA-iteration is

(3.2)' $W_{MLA} = [5+\frac{4}{3}(4.75+17*s)]\ N,$

where it is again assumed that the residuals are computed by means of the matrices $\tilde{R}^{\ell}$ for $\ell = 1(1)k$.

3.3 The computational work of the ILU-decompositions

The number of operations to perform the ILU-7 decomposition on Ω^k (cf.[9]) is 17N. Then the computational work of the ILU-7 decompositions on Ω^{ℓ} for $\ell = 1(1)k$ W_{IDEC} is

(3.3) $W_{IDEC} = \frac{4}{3} * 17N = 22\frac{2}{3}N$ for $k > 1$.

3.4 The evaluation of the function $(\Phi^k)^{(j-1)}$.

Let the number of operations to perform one Σ_n^k - evaluation (the right-hand side of (1.2)) be denoted by W_Σ. Then, one evaluation of $(\Phi^k)^{(j-1)}$ (occurring in (2.1)) W_E costs one f^k - evaluation (the right-hand side of (1.1)) plus $(W_\Sigma + 13)$ N operations. It should be noted that in the implementation of the Newton iteration (2.1) $(\Phi^k)^{(j-1)}$ is computed as:

$$(\Phi^k)^{(j-1)} = \Sigma_n^k + b_0\tau^\nu f^k(t,(y^k)^{(j-1)}) + [I^k - b_0\tau^\nu J^k](y^k)^{(j-1)} - (y^k)^{(j-1)},$$

because $I^k - b_0\tau^\nu J^k$ is evaluated instead of J^k.

4. NUMERICAL EXPERIMENTS

4.1 The numerical 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 are semi-discretized on a uniform grid Ω^k with mesh width h^k using standard symmetric differences. The grid Ω^ℓ has grid parameter $h^\ell = 2h^{\ell+1}$ for $\ell = k-1, k-2, \ldots, 0$.

The problems we chose are all of the form

$$\frac{\partial U}{\partial t} = d(t,x_1,x_2)\left[\frac{\partial^2 U^r}{\partial x_1^2} + \frac{\partial^2 U^r}{\partial x_2^2}\right] + \left[\frac{\partial U}{\partial x_1}\right]^s + \left[\frac{\partial U}{\partial x_2}\right]^s + v(t,x_1,x_2), \qquad (4.1)$$

$$0 \le t \le 1,$$

where the coefficient $d(t,x_1,x_2)$ and the term $v(t,x_1,x_2)$ are specified below, and the integer r and s are nonlinearity parameters. The Dirichlet boundary conditions as well as the initial condition at $t_0 = 0$ follow from the exact solution given in table 4.1.

Table 4.1. Specification of the test problems.

Example	Solution	$d(t,x_1,x_2)$	r	s	$v(t,x_1,x_2)$
I^a	$1 + de^{-t}(x_1^2+x_2^2)$	1	1	0	$-e^{-t}(x_1^2+x_2^2+4d) - 2$
I^b		100			
II	$1 + e^{-t}(x_1^2+x_2^2)$	$\frac{1}{1+t}$	1	2	$-e^{-t}[4d+(1+4e^{-t})(x_1^2+x_2^2)]$
III	$\frac{1}{2}(x_1+x_2)\sin 2\pi t$	$\frac{x_1+x_2}{2(1+t)}$	3	0	$-\left[\frac{3}{4}\frac{(x_1+x_2)^2}{1+t}\sin^3 2\pi t + 2 - \pi(x_1+x_2)\cos 2\pi t\right]$
IV	$\left[\frac{4}{5}(2t+x_1+x_2)\right]^{1/4}$	1	5	0	-2

4.2 The numerical scheme

In the case of 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 (BDF4) [7] which results in

$$(4.2) \qquad b_0 = \frac{12}{25}\,,\ \Sigma_n^k = \frac{1}{25}[48y_n^k - 36y_{n-1}^k + 16y_{n-2}^k - 3y_{n-3}^k]$$

in the iteration process (2.1).

In order to apply (4.2) four starting values are required which were obtained from the exact solution of the initial-boundary value problems. Furthermore, the Jacobian matrices J^{ℓ} for $\ell = 0(1)k$, were obtained by analytical differentiation. In case of nonlinear problems the Jacobian matrices were updated at the beginning of every integration step. In section 2 we have already specified the restriction, prolongation and the relaxation method (ILU-7) in the multi-level algorithm. The parameters p,q and s (see section 2) in MLA, the number of Newton steps per integration step m, the number of MLA-iterations per Newton step M and the predictor formula in (2.1) are still to be specified.

The variant MLA(k,p,1,s) appears to be more efficient than MLA(k,p,2,s), although MLA (k,p,2,s) has a slightly smaller convergence factor than MLA(k,p,1,s) (cf. [4,8]). Therefore, we choose q = 1 in MLA.

For the PMG method in $[E\{MLA(k,p,1,s)\}^M]^m$ mode (see section 2) the parameters p,s,k,M and m will be specified in the tables of results. In section 4.3 different variants of MLA(k,p,1,s) will be considered.

In [5] the *predictor formula* starting the iteration process (2.1) is

$$(4.3) \qquad (y^k)^{(pred)} = y_n^k.$$

Instead of the zero order predictor formula (4.3) we will also use in the PMG method the *third order extrapolation formula*

$$(4.4) \qquad (y^k)^{pred} = 4(y_n^k + y_{n-2}^k) - 6y_{n-1}^k - y_{n-3}^k$$

as predictor formula. The predictor formulas (4.3) and (4.4) are both asymptotically stable (cf.[8]). Therefore, on the ground of accuracy considerations it is obvious that the more accurate predictor formula (4.4) is to be preferred in the PMG method. In [10, Appendix B] the effect of the predictor formulas (4.3) and (4.4) in the PMG method is illustrated for

the examples III and IV. In the sections 4.3 and 4.4 we want to apply the PMG method for a large value of M, viz. M = 8. Using the predictor (4.4) for large values of M we can expect that the results are influenced by limiting precision difficulties. Therefore, the less accurate predictor (4.3) has been used in the sections 4.3 and 4.4. In section 4.5 we use the third order predictor (4.4).

In the tables of results we listed the accuracy measured by the number sd of correct significant digits at t = 1 defined by

$$(4.5) \qquad sd = \min_{\Omega^k} (-{}^{10}\log \mid \text{exact solution-numerical solution} \mid).$$

Furthermore, we use the notations:

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

$$(4.6) \qquad r_{av} = \left(\frac{\| (V^k)^{(M)} - (V^k)^{(M-1)} \|_2}{\| (V^k)^{(1)} - (V^k)^{(0)} \|_2} \right)^{1/M-1}, \quad M \geq 2,$$

where $\| \ \|_2$ is the Euclidean norm, $(V^k)^{(i)}$ is the i-th iterand of the MLA-iteration and $(V^k)^{(0)}$ is the starting value for the MLA-iteration in the j-th Newton iteration (2.1) given by $(V^k)^{(0)} = (y^k)^{(j-1)}$.

$W_{0.1}$ the number of operations to obtain a factor 0.1 reduction of the error by application of the multi-level algorithm, i.e.

$$(4.7) \qquad W_{0.1} = W_{MLA} \ / \ |{}^{10}\log r_{av}|,$$

where W_{MLA} is the number of operations of one MLA-iteration.

By numerical experiments we compute the values of r_{av} and $W_{0.1}$. Based on these numbers we determine what variants of the MLA are more efficient.

4.3 The effect of grid refinement and the parameters p and s

In table 4.2 we illustrate the effect of the parameters p and s in MLA and the effect of the grid parameter h^k on r_{av} (4.6) and $W_{0.1}$ (4.7) for problem I^b. For the linear problem I^b the starting values of the BDF4 are computed at t = $0,\tau,2\tau,3\tau$. We take just one integration step (i.e. $\tau=\frac{1}{4}$), because we are only interested in the solution of a linear system.

It is obvious that the effect of grid refinement and the parameters p and s is independent of the choice of the predictor formula in the PMG method. In the PMG method we choose m = 1 and M = 8 for problem I^b. In order to avoid the limiting precision difficulties we choose the zero order predictor formula (4.3).

Table 4.2. Results for problem I^b with $\tau = 1/4$, m = 1 and M = 8.

MLA-mode:		MLA(k,1,1,1)		MLA(k,1,1,0)		MLA(k,0,1,1)	
W_{MLA}:		$58\frac{1}{3}N$		$35\frac{2}{3}N$		34N	
k	h^k	r_{av}	$W_{0.1}$	r_{av}	$W_{0.1}$	r_{av}	$W_{0.1}$
2	1/20	0.022	35.2N	0.066	30.2N	0.072	29.8N
2	1/24	0.021	34.8N	0.067	30.4N	0.072	29.8N
3	1/32	0.023	35.6N	0.067	30.4N	0.074	30.1N
3	1/40	0.023	35.6N	0.066	30.2N	0.073	29.9N
3	1/48	0.022	35.2N	0.065	30.0N	0.072	29.8N

The results in table 4.2 show that the average reduction factor r_{av} of the MLA-algorithms are independent of the grid parameter h^k. The numbers of operations in one MLA(k,1,1,1) and MLA(k,1,1,0)-iteration are given by (3.2) and the number of operations in one MLA(k,0,1,1)-iteration is given by (3.2)'. For this problem the variant MLA(k,1,1,0) is more efficient than the variant MLA(k,1,1,0). Although MLA(k,0,1,1) is slightly more efficient than MLA(k,1,1,0) we prefer the variant MLA(k,1,1,0), because it is cheaper in storage (see Remark 2.1) and its average reduction factor of the error is smaller. Smaller values of r_{av} lead to more accurate numerical solutions. The numerical results in [10, Appendix B] show that for the accuracy smoothing before the coarse grid correction is preferable to smoothing after the coarse grid correction.

REMARK 4.1. Implementation of MLA(k,0,1,1) in a "space-economic" way results in more computational effort. In this case the residual $\phi^k - [I^k - b_0\tau^\nu J^k]x^k$ is determined by means of the original matrix, which results in 10N operations instead of 4N. Then, it is obvious that in this case MLA(k,0,1,1) is less efficient than MLA(k,1,1,0).

In [3] Hemker shows that in order to obtain a small error in a multi-level cycle pre-relaxation is to be preferred, while post-relaxation is preferred in a multi-level cycle to obtain a small residual.

4.4 Comparison of the PMG method with the PCGC method

The PCGC method is based on the same Newton-iteration (2.1), but now the systems of linear equations occurring in (2.1) are solved by the iterative use of a two-level algorithm (TLA). Notice that in TLA the grids Ω^k and Ω^{k-1} are only used. The two-level algorithm TLA consists of p ILU-7 relaxation sweeps on Ω^k, a coarse grid correction and again s ILU-7 relaxation sweeps on Ω^k. In the coarse grid correction we solve a discretized problem on Ω^{k-1} iteratively by means of μ-iterations with ILU-7 relaxation (cf.[9]). A particular two-level algorithm will now be denoted by TLA(p,μ,s). Furthermore, for the PCGC method which is also based on (2.1) we introduce the following notation $[E\{TLA(p,\mu,s)\}^M]^m$, where E and m are defined in section 2 and here M denotes the number of TLA(p,μ,s)-iterations per Newton step.

In this section we compare the PMG method in $[E\{MLA(k,1,1,1)\}^M]^m$ mode with the PCGC method in $[E\{TLA(1,4,1)\}^M]^m$ and $[E\{TLA(1,8,1)\}^M]^m$ mode. In the comparison we have to choose the same number of pre-and post-relaxations in both methods (here, $p = s = 1$). Other choices for the parameters p and s lead to the same conclusion. Further, the choice of the predictor formula is not so important here, because both methods are based on (2.1). For the same reasons as explained in the preceding subsections we choose the zero order predictor formula (4.3). For problem I^b the starting values of the BDF4 are computed at $t = 0,\tau,2\tau,3\tau$ with $\tau = \frac{1}{4}$ (i.e., just one integration step).

In figure 4.1 the average reduction factor r_{av} ((4.6) with $M = 8$) of the MLA-and TLA-algorithms are illustrated as a function of $1/(h^k)^2$ for the linear example I^b (i.e., $m = 1$ in both methods). In MLA(k,1,1,1) we put $k = 1$ for $h^k = 1/10, 1/12$, $k = 2$ for $h^k = 1/16, 1/20, 1/24$ and $k = 3$ for $h^k = 1/32, 1/40, 1/48$. The experimental values of r_{av} obtained by TLA (1,4,1), TLA(1,8,1) and MLA(k,1,1,1) are denoted by 0, ● and x, respectively.

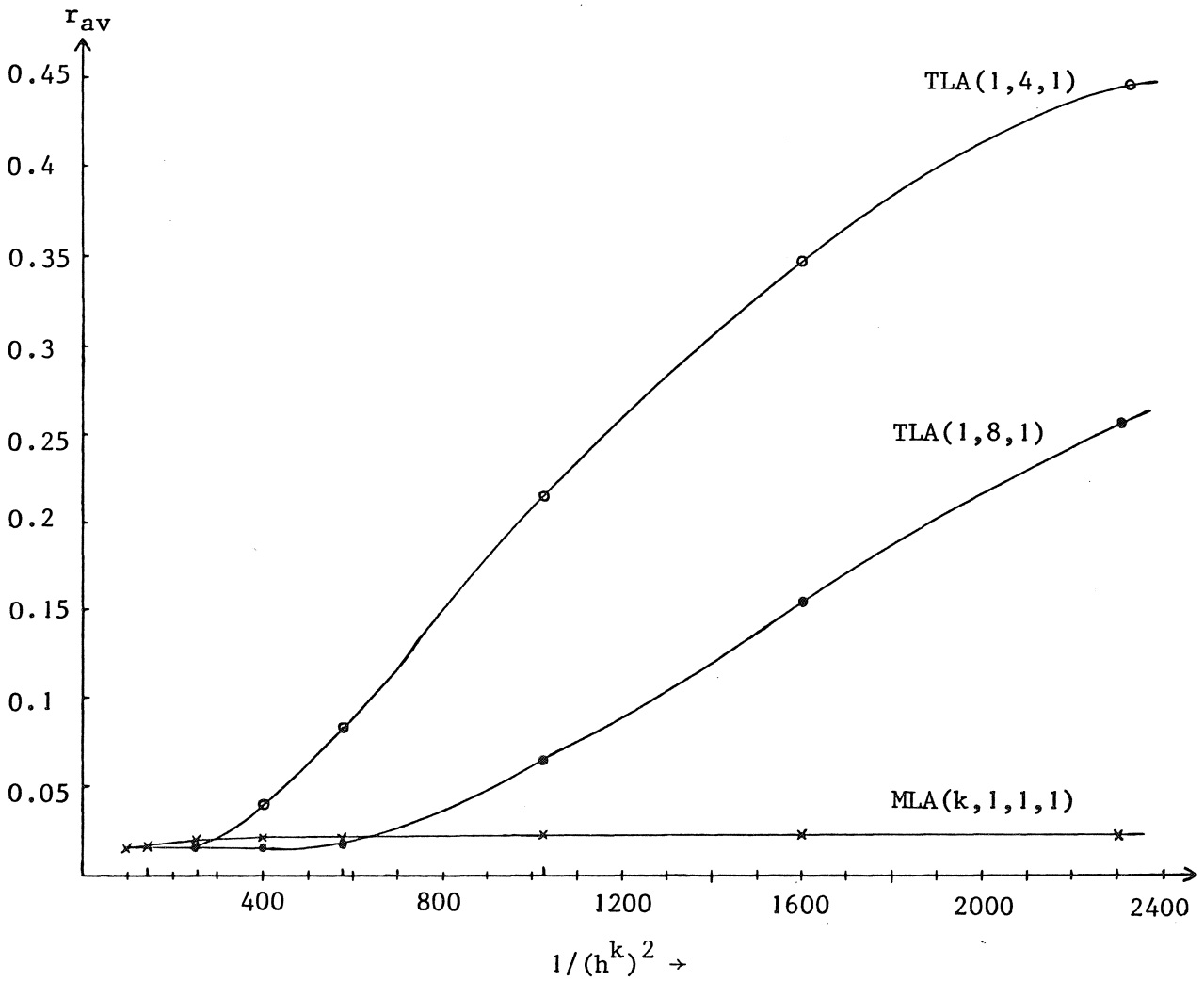

Fig. 4.1. The values of r_{av} ((4.6) with M = 8) as a function of $1/(h^k)^2$ obtained by TLA(1,4,1), TLA(1,8,1) and MLA(k,1,1,1) for problem I^b with $\tau = \frac{1}{4}$.

Figure 4.1 shows that for the range of h^k values the average reduction factor of MLA(k,1,1,1) remains more or less constant, whereas the average reduction factor of TLA(1,4,1) and TLA(1,8,1) increases if h^k decreases. In order to compare the efficiency of the PMG and PCGC method we have to consider only the computational work in the inner iteration (MLA-or TLA-iteration), because both methods are based on the same Newton iteration. For this linear example the decompositions and Jacobian evaluations are required only once. Therefore, the preliminary work can be neglected without

affecting the comparison. The number of operations of one MLA(k,1,1,1)-iteration (W_{MLA}) is atmost $58\frac{1}{3}N$, whereas the numbers of operations of one TLA(1,4,1)-and TLA(1,8,1)-iteration are (cf. [9]) 59.75N and 76.75N, respectively. Thus, for small h^k values the PMG method is more efficient than the PCGC method for problem I^b.

In table 4.3 the sd-values for problem I^b are illustrated obtained by $E[TLA(1,4,1)]^4$, $E[TLA(1,8,1)]^4$ and $E[MLA(k,1,1,1)]^4$ for a range of h^k-values. The PCGC method loses accuracy when the grid parameter h^k decreases. In the PMG method the accuracy remains more or less constant (sd≈4.8, being the accuracy of the BDF4 method for $\tau = 1/4$).

Table 4.3. sd-values for problem I^b with $\tau = 1/4$ obtained by $E[TLA(1,4,1)]^4$, $E[TLA(1,8,1)]^4$ and $E[MLA(k,1,1,1)]^4$.

k	h^k	$E[TLA(1,4,1)]^4$	$E[TLA(1,8,1)]^4$	$E[MLA(k,1,1,1)]^4$
1	1/10	4.71	4.71	4.71
1	1/12	4.72	4.72	4.72
2	1/20	4.83	4.70	4.76
2	1/24	3.19	4.71	4.75
3	1/32	1.51	3.68	4.77
3	1/40	.65	2.09	4.76
3	1/48	.16	1.19	4.77

Additional experiments have shown that for the nonlinear example IV with $\tau = 1/10$, $h^k = 1/32$, $k = 3$, $m = 1,2,3,4$ the PMG method in $[E(MLA(k,1,1,1))]^m$ mode is more efficient than the PCGC method in $[E(TLA(1,4,1))]^m$ and $[E(TLA(1,8,1))]^m$ mode. In both methods the zero order predictor formula (4.3) has been used. The computational work in the PMG method is more or less equal to the computational work in the $[E(TLA(1,4,1))]^m$ method, but the PMG method is more accurate. The PMG method

and the $[E(TLA(1,8,1))]^m$ method produce the same accuracy for the same value of m, however the PMG method is cheaper.

For nonlinear problems the number of operations per integration step to perform the ILU-decompositions in the PCGC method and the PMG method is equal to 21.25N and $22\frac{2}{3}$N, respectively. The computational work of the evaluation of the Jacobian matrices per integration step in the PCGC method and the PMG method is equal to $\frac{5}{4}$ W_J and $\frac{4}{3}$ W_J, respectively, where W_J denotes the computational work of the evaluation of $I^k - b_0 \tau^\nu J^k$.

To implement the PCGC method we need $23\frac{1}{2}$ arrays of length N for storage. The PMG method requires the storage of $I^\ell - b_0 \tau^\nu J^\ell$ for $\ell = 0(1)k$, $\tilde{L}^\ell$, $\tilde{U}^\ell$ and $\tilde{R}^\ell$ for $\ell = 1(1)k$, L^0, U^0, $(\Phi^k)^{(j-1)}$ and y^k_{n+1-s} for $s = 0(1)4$ when the BDF4 method (4.2) is chosen as implicit formula (1.2). Then to implement the PMG method with ILU-7 relaxation we need at most $24\frac{2}{3}$ arrays of length N for storage.

The final conclusion (based on the experiments) is that for a small grid parameter h^k the PMG method is considerably more efficient than the PCGC method when also the preliminary work and the storage requirements are taken into account.

4.5. Comparison of the PMG method with the SC method

From the results of the preceding subsections and [10, Appendix] it follows that the PMG method in $E[MLA(k,1,1,0)]^2$ mode with the third order predictor formula (4.4) appears to be a good choice. This particular PMG method has been compared with the SC method described in [6]. Notice that in this PMG method one Newton iteration ((2.1) with m=1) is sufficient in order to obtain a fourth order accurate method (see Section 2).

The SC method is also based on BDF4 (4.2). In the SC method the system of equations (1.2) is solved by a splitting method (ADI) and this iteration process is accelerated by using Chebyshev polynomials. The initial approximation used in the iteration process is the third order extrapolation formula (4.3) smoothed by an adjusted Jacobi iteration. The resulting fourth order, fourstep splitting method (called SC method) has a real stability boundary bounded below by $c\tilde{m}^4$, $\tilde{m}$ being the number of iterations and the constant c is approximately equal to 4. In the SC method we performed only

one Newton iteration in solving each implicit relation.

In the PMG method the initial-boundary value problems were semi-discretized using standard differences on the uniform grids Ω^{ℓ}, $\ell = 0(1)k$ with $h^k = 1/24$ and $k = 3$. In the SC method we only use one grid, viz. Ω^k.

Our test examples are problem II and III both with $h^k = 1/24$ and $k = 3$. For these examples the starting values needed by the PMG method and the SC method were obtained by computing them from the exact values prescribed at $t = -3\tau, -2\tau, -\tau, 0$.

In order to compare both methods we introduce the following notations:

- Σf total number of f^k-evaluations (the right-hand side of (1.1))
- ΣJ total number of Jacobian evaluations on Ω^k for the SC method and on Ω^{ℓ} with $\ell = 0(1)k$ for the PMG method
- ACE the additional computational effort expressed in the total number of operations on Ω^k for the SC method and on Ω^{ℓ} with $\ell = 0(1)k$ for the PMG method.

A conclusion based on the sd- and Σf- values as to which method is the more efficient one is difficult, since on should also measure the additional computational effort required by both methods. Therefore, we list in the tables of results also the ACE-values required by the PMG method and the SC method. For a nonlinear problem the value of ACE for the particular PMG method used in this section is given by $\frac{1}{\tau} * 119N$ operations and the value of ACE for the SC method is given by $[\frac{16}{\tau} + 15 * \Sigma\, I_{CHEB}]N$ operations, where N is the number of inner points of the grid Ω^k and $\Sigma\, I_{CHEB}$ denotes the total number of Chebyshev iterations in the SC method. The Σf-values in the PMG method and the SC method are given by $\frac{m}{\tau}$ and $2 * \Sigma I_{CHEB} + \frac{1}{\tau}$, respectively. For a more detailed discussion of the additional computational effort in the PMG and SC method we refer to [10, Appendix C].

It should be noted that for nonlinear problems the computational work of the evaluation of the Jacobian matrices in the PMG method is equal to $\tau^{-1} * \frac{4}{3} * W_J$, where W_J denotes the computational work of the evaluation of $I^k - b_0\, \tau^{\nu}\, J^k$.

Table 4.4a. Results for problem II with $h^k = 1/24$ obtained by $E[MLA(3,1,1,0)]^2$ with the predictor (4.4).

τ	sd	Σf	ΣJ	ACE
1/5	4.8	5	5	595N
1/10	6.08	10	10	1190N
1/20	7.34	20	20	2380N
1/40	8.57	40	40	4760N

Table 4.4b. Results for problem II with $h^k = 1/24$ obtained by the SC method.

τ	sd	Σf	ΣJ	ACE
1/5	*	-	-	-
1/10	*	-	-	-
1/20	6.10	140	20	1220N
1/40	7.54	212	40	1930N
1/80	8.71	400	80	3680N

In the tables 4.4a and 4.4b numerical results are listed for problem II obtained by the PMG and SC method. An asterisk indicates instability. In order to produce the same accuracy the SC method requires considerably more f^k-evaluations. For $\tau = 1/5$ and $\tau = 1/10$ the PMG method is already very accurate, whereas the SC method is unstable. For small τ-values the additional computational effort in the SC method is less than in the PMG method.

Table 4.5a. Results for problem III with $h^k = 1/24$ obtained by $E[MLA(3,1,1,0)]^2$ with the predictor (4.4).

τ	sd	Σf	ΣJ	ACE
1/10	*	-	-	-
1/15	2.72	15	15	1785N
1/20	3.25	20	20	2380N
1/40	4.3	40	40	4760N
1/80	5.45	80	80	9520N
1/160	6.64	160	160	19040N

Table 4.5b. Results for problem III with $h^k = 1/24$ obtained by the SC method.

τ	sd	Σf	ΣJ	ACE
$\geq 1/40$	*	-	-	-
1/80	5.89	390	80	3605N
1/160	6.89	676	160	6430N

In the tables 4.5a and 4.5b the numerical results are listed obtained by the PMG and SC method for problem III. For $\tau \geq 1/40$ the SC method failed because the Newton process did not converge (indicated by *), whereas the PMG method failed only for $\tau = 1/10$ because of divergence of the Newton process. The SC method requires again considerably more f^k-evaluations than the PMG method. However, for $\tau = 1/80$ and 1/160 the additional computational effort in the SC method is considerably less than in the PMG method. If a Jacobian - vector operation is not cheap in comparison with a f^k-evaluation, the SC method becomes competitive with the PMG method for small τ-values.

The fourth order behaviour of the PMG method used in this section is more or less reflected in the sd-values for the problems II and III (on halving the integration step the sd-values should increase with $4^{10}\log 2 \simeq 1.2$).

From the tables of results we may draw the following conclusions:

(i) For large integration steps the PMG method is more efficient and robust than the SC method. Moreover, the SC method is sensitive to grid refinement [5].

(ii) For small integration steps the SC method becomes competitive to the PMG method, if the spectral radius of $\partial f^k/\partial y^k$ is not too large and if a f^k-evaluation is not (extremely) expensive.

Additional experiments have shown (see [10, Appendix C]) that for the examples I^a and IV the PMG method is also superior to the SC method. Furthermore, one should also take into account that the SC method requires 15 arrays of length N for storage, whereas the PMG method requires approximately 25 arrays of length N for storage (see Section 4.4).

5. CONCLUDING REMARKS

The experiments reported in the preceding sections show the superiority of the PMG method over the SC method and the PCGC method. The PMG method is more robust and is insensitive to grid refinement, whereas the other methods loose accuracy if the grid parameter h^k is decreased. The PMG method with the third order extrapolation formula (4.4) as predictor formula shows its fourth order behaviour for realistic integration steps.

From the experiments reported in [10, Appendix A] it appears that for problem I^b the ILU-7 relaxation is more efficient than the ILU-5 and ILU-9 relaxation. For the accuracy smoothing before the coarse grid correction is preferable to smoothing after the coarse grid correction.

If the aspect of storage is as important as the computational effort than explicit integration formulas should be considered which are cheaper in storage but more expensive in computational effort than the PMG method. The number of arrays for storage can be reduced by choosing a lower order BDF formula and other relaxation methods (e.g., ILU-5 relaxation and Point Gauss-Seidel iteration) in the PMG method. Then, however the PMG method becomes less efficient.

In the PMG method described in this paper there are several choices which are not necessarily the best possible. For instance the 9-point re-

striction and 9-point prolongation may be replaced by the 7-point restriction and 7-point prolongation (cf. [4,8]) which makes the PMG method slightly more efficient. Futhermore, on the coarsest grid Ω^0 it is also possible to solve the linear equations approximately by some iterative method instead of exactly. In order to construct an optimal MLA-algorithm the number of levels should also be considered. Mostly one chooses the coarsest grid-size h^0 as coarse as possible.

ACKNOWLEDGEMENT

The author would like to thank Mr. B.P. Sommeijer for his constructive remarks and careful reading of the manuscript.

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., *A note on defect correction processes with an approximate inverse of deficient rank*, J. Comp. Appl. Math. 8 (1982), pp. 137-139.

[4] HEMKER, P.W., *On the comparison of line-Gauss-Seidel and ILU-relaxation in multigrid algorithms*, Report NW 129/82, Mathematisch Centrum, Amsterdam (1982).

[5] HOUWEN, P.J. van der & H.B. de VRIES, *Preconditioning and coarse grid corrections in the solution of the initial value problem for nonlinear partial differential equations*, SIAM J. on Sci. and Stat. Comp. 3 (1982), pp. 473-485.

[6] HOUWEN, P.J. van der & H.B. de VRIES, *A fourth order ADI method for semidiscrete parabolic equations*, Report NW 128/82, Mathematisch Centrum, Amsterdam (1982) (to appear in J. Comp. Appl. Math. 9

(1983)).

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

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

[9] VRIES, H.B. de, *The two-level algorithm in the solution of the initial value problem for partial differential equations*, Report NW 136/82, Mathematisch Centrum, Amsterdam (1982).

[10] VRIES, H.B. de, *The multi-grid method in the solution of time-dependent nonlinear partial differential equations*, Report NW 148/83, Mathematisch Centrum, Amsterdam (1983).

APPENDIX

In this appendix several additional experiments are reported. With the exception of appendix A we use only ILU-7 relaxation in the multi-level algorithm.

In appendix A numerical results of the MLA(k,1,1,1)-algorithm with ILU-5, ILU-7 and ILU-9 relaxation are listed for problem I^b.

In appendix B the effect of the predictor formulas (4.3) and (4.4) in the PMG method is illustrated by numerical experiments with the examples III and IV.

Finally, in appendix C we give a more detailed discussion of the additional computational effort in the PMG method and the SC method. For the examples I^a and IV we give also numerical results obtained by the PMG method and SC method.

For more details concerning the implementation and the notation we refer to the preceding sections.

A. Numerical results obtained by MLA with three different ILU-relaxations

Firstly, we consider the computational work of one MLA-iteration (W_{MLA}) when we choose ILU-5 and ILU-9 relaxation as relaxation method in MLA. In section 3 W_{MLA} is derived when ILU-7 relaxation is used in MLA.

For the ILU-5 and ILU-9 relaxation on the grid Ω^k the number of operations is equal to 13N and 25N, respectively (cf. [9]).

When the residual $\phi^k - [I^k - b_0\tau^\nu J^k]\, x^k$ is determined by means of the matrix $\tilde{R}^k$ (see Remark 2.1) the numbers of operations are:

using ILU-5 relaxation in MLA : 4N

using ILU-9 relaxation in MLA : 8N.

Then, the computational work to perform one MLA(k,p,q,s)-iteration is

$$W_{MLA} = \frac{4}{4-q}\left[9.75 + (p+s)13\right]N \quad \text{for ILU-5,} \tag{A.1}$$

$$W_{MLA} = \frac{4}{4-q}\left[13.75 + (p+s)25\right]N \quad \text{for ILU-9.} \tag{A.2}$$

As in section 3 we assume $k > 1$ (more than two grids) and $q < 4$.

The number of operations to perform the ILU-5 and ILU-9 decomposition on Ω^k (cf. [9]) is 8N and 28N, respectively. Then the computational work of the ILU-decompositions on Ω^ℓ for $\ell = 1(1)k$ (W_{IDEC}) is:

(A.3) $$W_{IDEC} = 10\tfrac{2}{3}N \quad \text{for} \quad \text{ILU-5},$$

(A.4) $$W_{IDEC} = 37\tfrac{1}{3}N \quad \text{for} \quad \text{ILU-9}.$$

For the ILU-5, ILU-7 and ILU-9 relaxation the numbers of arrays of length N to store $\tilde{L}^\ell$, $\tilde{U}^\ell$ and $\tilde{R}^\ell$ for $\ell = 1(1)k$ are $9\frac{1}{3}$, 12 and $17\frac{1}{3}$, respectively.

Table A.1. Results obtained by MLA(k,1,1,1) with ILU-5, ILU-7 and ILU-9 relaxation for problem I^b with $\tau = \frac{1}{4}$, $m = 1$ and $M = 7$.

Method:	MLA(k,1,1,1) with ILU-5		MLA(k,1,1,1) with ILU-7		MLA(k,1,1,1) with ILU-9	
W_{MLA}:	$47\frac{2}{3}N$		$58\frac{1}{3}N$		85N	
h^k k	r_{av}	$W_{0.1}$	r_{av}	$W_{0.1}$	r_{av}	$W_{0.1}$
1/20 2	0.052	37.1N	0.022	35.2N	0.014	45.8N
1/24 2	0.05	36.6N	0.021	34.8N	0.014	45.8N
1/32 3	0.056	38.1N	0.023	35.6N	0.015	46.6N
1/40 3	0.053	37.4N	0.023	35.6N	0.015	46.6N

In table A.1 the results obtained by MLA(k,1,1,1) with ILU-5, ILU-7 and ILU-9 relaxation are listed for problem I^b on Ω^k for a range of h^k values. The starting values of the BDF4 are computed at $t = 0, \tau, 2\tau, 3\tau$. The results in table A.1 illustrate that the ILU-7 relaxation is to be preferred in MLA. Further, the ILU-5 relaxation is more efficient than the ILU-9 relaxation in MLA.

B. The predictor formula in the PMG method

In this section the effect of the predictor formulas (4.3) and (4.4) in the PMG method is illustrated by numerical experiments with the examples III and IV. The starting values of the BDF4 are computed at $t = -3\tau, -2\tau, -\tau, 0$ for problem III and at $t = 0, \tau, 2\tau, 3\tau$ for problem IV. Notice that for problem IV the exact solution does not allow to choose the starting values at $t = -3\tau, -2\tau, -\tau, 0$.

Table B.1. Results for problem III with $h^k = 1/24$ and $\tau = 1/20$ obtained by $[E\{MLA(3,p,1,s)\}^M]^m$ with (4.3) and (4.4).

Predictor	m	M	p = s = 1	p = 1, s = 0	p = 0, s = 1
(4.3)	1	1	.57	.57	.57
	2	2	*	*	*
(4.4)	1	1	3.23	3.06	3.01
	2	2	3.24	3.24	3.24

Table B.2. Results for problem IV with $h^k = 1/24$ and $\tau = 1/10$ obtained by $[E\{MLA(3,p,1,s)\}^M]^m$ with (4.3) and (4.4).

Predictor	m	M	p = s = 1	p = 1, s = 0	p = 0, s = 1
(4.3)	1	1	2.89	2.95	2.80
		2	2.87	2.87	2.87
	2	1	3.79	3.84	3.69
		2	3.77	3.77	3.77
(4.4)	1	1	5.43	4.20	4.03
		2	5.50	5.49	5.48
	2	1	6.63	6.17	6.08
		2	6.73	6.73	6.72

In the tables B.1 and B.2 the results are listed obtained by the PMG method for the examples III and IV, respectively. An asterisk indicates instability. From the tables of results the following conclusions can be drawn:

(i) The third order predictor formula (4.4) is to be preferred in the PMG method.

(ii) For the accuracy smoothing before the coarse grid correction is preferable to smoothing after the coarse grid correction.

C. The additional computational effort in the PMG method and the SC method.

Here we give a more detailed discussion of the additional computational effort of the PMG method (in $\{E[MLA(3,1,1,0)]^M\}^m$ mode with the predictor formula (4.4)) and the SC method. For the examples I^a and IV numerical results are also shown obtained by the PMG method and the SC method.

The additional computational effort (ACE) of the PMG method is given by the total number of operations in the PMG method on the grid Ω^ℓ (ℓ=0(1)k) required for the decompositions, the MLA-iterations, the calculations of the predictor (4.4) and the evaluations of $(\Phi^k)^{(j-1)}$ with the exception of the f^k-evaluations (see also section 3). The number of operations to compute Σ_n^k given by (4.2) (W_Σ) and the predictor formula (4.4) (W_{PRED}) is 7N and 5N, respectively. The number of operations in one MLA(3,1,1,0)-iteration (W_{MLA}) is $35\frac{2}{3}$N and the number of operations to perform the ILU-7 decompositions on Ω^ℓ for ℓ = 1(1)k (W_{IDEC}) is $22\frac{2}{3}$N. Then for a nonlinear problem the value of ACE of the PMG method described in Section 4.5 is given by

$$\text{(C.1)} \qquad ACE = \frac{1}{\tau} * \left[m * M * 35\frac{2}{3} + m * 20 + 27\frac{2}{3}\right]N.$$

Choosing m = 1 and M = 2 we obtain $ACE = \frac{1}{\tau} * 119N$.
For a linear problem ACE is given by

$$\text{(C.2)} \qquad ACE = [\frac{1}{\tau} * (M*35\frac{2}{3} + 25) + 22\frac{2}{3}]N,$$

because the ILU-decompositions are required once and m = 1 in the PMG method.

The additional computational effort of the SC method is given by the

total number of operations in the SC method on the grid Ω^k required for the decomposition of tridiagonal matrices, the solution of tridiagonal systems, the Chebyshev iterations, the calculations of the predictor formula (4.4) and the right-hand side of (1.2) (Σ_n^k). In the SC method the evaluation of the spectral radius of the Jacobian matrix $\partial f^k/\partial y^k$ and all initial work for estimating the iteration parameters are neglected. Furthermore, in the Jacobi iteration only the f^k-evaluation is taken into account and in the splitting process only the f^k-evaluations, solution of tridiagonal systems and the decomposition of tridiagonal matrices are taken into account. The number of operations to perform a LU-decomposition of a tridiagonal matrix and the number of operations to solve a tridiagonal system of linear equations is 3N and 5N, respectively. The number of operations required for the Chebyshev iterations is $[5 * \Sigma I_{CHEB} - \frac{2}{\tau}]N$, where ΣI_{CHEB} denotes the total number of Chebyshev iterations. Then for a nonlinear problem the value of ACE of the SC method is given by

$$\text{(C.3)} \qquad ACE = [\frac{16}{\tau} + 15 * \Sigma I_{CHEB}]N.$$

For a linear problem ACE is given by

$$\text{(C.4)} \qquad ACE = [\frac{10}{\tau} + 6 + 15 * \Sigma I_{CHEB}]N,$$

because the LU-decompositions of the tridiagonal matrices are required once.

Table C.1a. Results for problem I^a with $h^k = 1/24$ obtained by $E[MLA(3,1,1,0)]^M$ with (4.4).

τ	M	sd	Σf	ACE
1/5	1	3.84	5	326N
	2	4.94	5	$504\frac{1}{3}$N
1/10	1	5.24	10	$629\frac{1}{3}$N
	2	6.24	10	986N
1/20	1	6.45	20	1236N
	2	7.51	20	$1949\frac{1}{3}$N
1/40	1	7.70	40	$2449\frac{1}{3}$N
	2	8.78	40	3876N

Table C.1b. Results for problem I^a with $h^k = 1/24$ obtained by the SC method

τ	ΣI_{CHEB}	sd	Σf	ACE
1/5	20	4.0	45	356N
1/10	40	5.13	90	706N
1/20	60	6.3	140	1106N
1/40	120	7.43	280	2206N
1/80	160	8.73	400	3206N

In the tables C.1a and C.1b numerical results for the linear example I^a are listed obtained by the PMG method and the SC method. The starting values for BDF4 are computed at $t = -3\tau, -2\tau, -\tau, 0$. Notice that in both methods the Jacobians are determined once. The SC method requires considerably more f^k-evaluations than the PMG method. For the same value of τ the PMG method in E[MLA(3,1,1,0)] mode produces more or less the same accuracy as the SC method. In this case for $\tau = 1/5$ and $\tau = 1/10$ the additional computational effort in the PMG method is even less than in the SC method. Two MLA(3,1,1,0)-iterations in the PMG method is to be preferred.

Table C.2. Results for problem IV with $h^k = 1/24$ obtained by $E[MLA(3,1,1,0)]^2$ with (4.4).

τ	sd	Σf	ΣJ	ACE
1/10	5.89	10	10	1190N
1/20	7.29	20	20	2380N
1/40	7.49	40	40	4760N

In table C.2 numerical results for problem IV are listed obtained by the PMG method. For example IV the estimate of the spectral radius of $\partial f^k/\partial y^k$ used in the SC method is given by $64(1+t)/(h^k)^2$. For $\tau = 1/10$, 1/20 and 1/40 the SC method was unstable. For $\tau = 1/80$ we obtained by the SC

method the following results:

$$sd = 7.52,\ \Sigma f = 720,\ \Sigma J = 80 \text{ and } ACE = 6080N.$$

The starting values of the BDF4 are computed at $t = 0, \tau, 2\tau, 3\tau$ for the same reason as explained in appendix B. Although in the numerical experiments with the PMG method and the SC method the number of integration steps is $(\tau^{-1}-3)$ for this example, we assume for convenience that in the comparison of both methods the number of integration steps is τ^{-1}. For this example the PMG method is superior to the SC method. The number of Chebyshev iterations ($\Sigma I_{CHEB}=320$) in the SC method is considerably even for $\tau = 1/80$. This is due to the large value of the spectral radius. As a consequence the SC method is rather expensive for this example.

The results in table C.2 indicate that the asymptotic order 4 of the PMG method is not shown. The explanation is the effect of the space discretization error.