stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 95/80 DECEMBER

P.J. VAN DER HOUWEN & H.B. DE VRIES

PRECONDITIONING AND COARSE GRID CORRECTIONS IN THE SOLUTION OF THE INITIAL VALUE PROBLEM FOR NONLINEAR PARTIAL DIFFERENTIAL EQUATIONS

Preprint

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

Preconditioning and coarse grid corrections in the solution of the initial value problem for nonlinear partial differential equations *)

by

P.J. van der Houwen & H.B. de Vries

ABSTRACT

The numerical solution is discussed of nonlinear, time-dependent partial differential equations. By the method of lines an initial value problem for a system of ODE's is obtained to which an implicit linear multistep method is applied. Using Newton iteration the nonlinear implicit relations are replaced by a sequence of linear equations. The linear problems are preconditioned by applying incomplete LU-decomposition and then solved by Jacobi iteration. The convergence is accelerated by introducing coarse grid corrections. 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, incomplete LU-decomposition, coarse grid corrections*

---

*) This report will be submitted for publication elsewhere.

1. INTRODUCTION

When the method of lines is applied to an initial-boundary value problem for a (nonlinear) hyperbolic or parabolic differential equation, we often obtain a system of ODE's of the form

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

with prescribed values for y (and dy/dt) at $t = t_0$. By applying a linear multistep method to this equation we are asked to solve at each time step the system of equations

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

where y_n denotes the numerical solution at $t = t_n$, $\tau_n = t_{n+1} - t_n$ and $\{a_\ell, b_\ell\}$ are real coefficients. The (approximate) solution of (1.2) is identified with y_{n+1}.

Assuming that f is differentiable with respect to y we may define the iteration process [5]

$$\begin{aligned}
&y^{(0)} = y^{(\mathrm{pred})},\\
&y^{(j)} = [I - b_0\tau_n^{\nu}\tilde{J}_j]^{-1}\,[b_0\tau_n^{\nu}(J-\tilde{J}_j)y^{(j-1)} + \phi(u(j))], \qquad j = 1,\ldots,M\\
&\phi(u(j)) = \textstyle\sum_n + b_0\tau_n^{\nu}[f(t_{n+1}, y^{(u(j))}) - Jy^{(u(j))}],\\
&J = \frac{\partial f}{\partial y}(t_{n+1}, y^{(0)}), \qquad 0 \le u(j) \le u(j+1) \le j,
\end{aligned} \tag{1.3}$$

where $\sum_n$ denotes the right-hand side of equation (1.2), $y^{(\mathrm{pred})}$ is some predictor, $\tilde{J}_j$, $j = 1,\ldots,M$, are approximations to J and u(j) is a piecewise constant function. This function will be called the *update function* because each time it changes its value a new right-hand side function is to be evaluated.

The scheme (1.3) contains several well-known iteration processes as special cases. For instance,

$$\tilde{J}_j = J, \qquad u(j) = j-1 \tag{1.4}$$

yields the *modified Newton-Raphson process*.

A second class of methods is based on a *splitting of the Jacobian matrix* J. Well known methods arise if (f(t,y) originating from a 2-dimensional problem)

$$\tilde{J}_j = \begin{cases} J_1 & j \text{ odd} \\ J_2 & j \text{ even} \end{cases}, \quad J_1 + J_2 = J, \tag{1.5}$$

where J_1 and J_2 are "simply structured matrices" (e.g. tridiagonal matrices). A slight modification of (1.5) is given by

$$\tilde{J}_j = J_1 + J_2 - b_0\tau_n^{\nu} J_1 J_2 = J - b_0\tau_n^{\nu} J_1 J_2. \tag{1.5'}$$

The update function u(j) is still free in these methods. In [4] the case u(j) = j-1 was investigated (*method of successive corrections*). This choice requires in each iteration an f-evaluation and generally is rather expensive; however, the efficiency might be improved by choosing other update strategies for f.

A third class of iteration methods is based on *incomplete LU-decomposition*. Let L^*U^* denote an incomplete LU-decomposition of the matrix $I - b_0\tau^{\nu}J$ (from now on the index n is omitted in the steps τ_n), i.e.

$$I - b_0\tau^{\nu}J = L^*U^* - R, \tag{1.6}$$

where R is the residual matrix with a small matrix norm and L^*, U^* are a lower and upper triangular matrix, respectively. These matrices were chosen as proposed in [8]. Let J be a (K×K) matrix, then writing $A = I - b_0\tau^{\nu}J$ and denoting the elements of the matrices A, L^*, U^* and R by a_{ij}, ℓ^*_{ij}, u^*_{ij} and r_{ij}, $1 \le i,j \le K$, the incomplete LU-decomposition is defined by

$$\begin{aligned}
&\ell^*_{jj} = 1, \; j = 1,\ldots,K, \\
&\text{If } (k,j) \in P \text{ then } u^*_{kj} = 0 \Rightarrow r_{kj} := -\Big(a_{kj} - \sum_{i=1}^{k-1} \ell^*_{ki}u^*_{ij}\Big) \\
&\qquad \text{else } u^*_{kj} := a_{kj} - \sum_{i=1}^{k-1} \ell^*_{ki}u^*_{ij} \text{ for } j = k,\ldots,K; \\
&\text{If } (j,k) \in P \text{ then } \ell^*_{jk} = 0 \Rightarrow r_{jk} := -\Big(a_{jk} - \sum_{i=1}^{k-1} \ell^*_{ji}u^*_{ik}\Big) \\
&\qquad \text{else } \ell^*_{jk} := \Big(a_{jk} - \sum_{i=1}^{k-1} \ell^*_{ji}u^*_{ij}\Big)/u^*_{kk} \text{ for } j = k+1,\ldots,K,
\end{aligned} \tag{1.7}$$

where k = 1,...,K. In all our experiments P is the set of pairs of integers defined by

$$P = \{(i,j) \mid |i-j| \neq 0,1, \frac{b-3}{2}, \frac{b-1}{2}; 1 \leq i, j \leq K\},$$

where b denotes the bandwidth of A [11].

This choice of the set P is suitable when the partial differential equations does not contain mixed derivatives and is semi-discretized by standard symmetric differences. For more details on the L^*U^*-decomposition we refer to [8].

We now define the approximations $\tilde{J}_j$, j = 1,...,M in (1.3) by

$$(1.8) \qquad I - b_0\tau^\nu \tilde{J}_j = L^*U^*, J_j = \frac{1}{b_0\tau^\nu}[I - L^*U^*].$$

Substitution into (1.3) yields for $y^{(j)}$ the expression

$$(1.9) \qquad y^{(j)} = [L^*U^*]^{-1}[Ry^{(j-1)} + \phi(u(j))].$$

This iteration method can be interpreted as a Newton-Raphson method in which the linear systems are first preconditioned and then solved by Jacobi iteration. To see this we consider the linear system to be solved in the Newton-Raphson process for (1.2) in the form

$$(1.10) \qquad (I - b_0\tau^\nu J)y = \phi(u(j)),$$

where $y^{(u(j))}$ is the solution of the preceding Newton step. Let L^*U^* be the incomplete LU-decomposition defined by (1.7) then (1.10) can be preconditioned to obtain

$$(L^*U^*)^{-1}(I - b_0\tau^\nu J)y = (L^*U^*)^{-1}\phi(u(j))$$

Substitution of (1.6) yields

$$(1.10') \qquad y = [L^*U^*]^{-1}[Ry + \phi(u(j))]$$

and applying Jacobi iteration leads to (1.9).

In Section 2 we derive the iteration error of the general iteration method (1.3) and in section 3 the effect is considered of introducing coarse grid corrections into (1.3). Finally, in section 4 we apply (1.9) in a number of parabolic initial-boundary value problems and show that coarse grid corrections improve the accuracy considerably. Also comparisons are given with other integration techniques.

2. THE ITERATION ERROR

Let η be the solution of equation (1.2) and define the iteration error

$$\varepsilon(j) = \eta - y^{(j)}. \tag{2.1}$$

Then, it is easily verified that the iteration error of the scheme (1.3) satisfies the relation

$$[I - b_0\tau^\nu \tilde{J}_j]\,\varepsilon(j) = b_0\tau^\nu \Big\{[J - \tilde{J}_j]\varepsilon(j-1) - (J\eta - f(t_{n+1},\eta)) + (Jy^{(u(j))} - f(t_{n+1},y^{(u(j))}))\Big\}. \tag{2.2}$$

Since f is assumed to be differentiable it satisfies an inequality of the form

$$\|f(t,v) - Jv - f(t,w) + Jw\| \le \sup_{y\in Y} \Big\| \frac{\partial f}{\partial y}(t,y) - J\Big\| \, \|v-w\|, \tag{2.3}$$

where

$$Y = \{y \mid y = \Theta v + (1-\Theta)w, \quad 0 \le \Theta \le 1\}.$$

Using this inequality we derive from (2.2) the estimate

$$\begin{aligned} &\|\varepsilon(j)\| \le |b_0|\tau^\nu \|(I-b_0\tau^\nu \tilde{J}_j)^{-1}\| \; \{\|J - \tilde{J}_j\| \, \|\varepsilon(j-1)\| + C_j\|\varepsilon(u(j))\|\} \\ &C_j = \sup_{y\in Y_j} \Big\|\frac{\partial f}{\partial y}(t_{n+1},y) - \frac{\partial f}{\partial y}(t_{n+1},y^{(0)})\Big\|, \\ &Y_j = \{y \mid y = \Theta\eta + (1-\Theta)y^{(u(j))}, 0 \le \Theta \le 1\}. \end{aligned} \tag{2.4}$$

From (2.4) it is immediate that the final iteration error

$$\varepsilon(M) = O(\tau^{\nu(\hat{p}+1+m)}) \qquad \text{as } \tau \to 0 \tag{2.5}$$

where $\hat{p}$ is the order of accuracy of the predictor formula used in (1.3) and m denotes the number of f-evaluations. Thus in order to obtain the same order of accuracy p as the generating k-step method (1.2) at least $(p+1)/\nu - \hat{p} - 1$ right-hand side evaluations are required.

We also conclude from (2.4) that for slowly varying Jacobian matrices $\partial f/\partial y$ the contribution of the (Newton) error $\varepsilon(u(j))$ will be small when compared with $\varepsilon(j-1)$. This means that one should keep u(j) sufficiently long on a fixed value in order to compensate the possibly large error constant caused by the factor $\|J-\tilde{J}_j\|$. This may lead to large numbers of matrix-vector multiplications but does not increase the number of function evaluations.

3. COARSE GRID CORRECTION

In order to accelerate the convergence of (1.3) we add to $y^{(j-1)}$ for $j = M_1, M_2, \ldots$ the correction term (cf.[2,3])

$$c^{(j)} = P_{hH}[I-b_0\tau^{\nu}J_H]^{-1} R_{Hh}[\phi(u(j-1)) - (I-b_0\tau^{\nu}J)y^{(j-1)}] \tag{3.1}$$

to obtain

$$y^{(j)} = [I-b_0\tau^{\nu}\tilde{J}_j]^{-1} [b_0\tau^{\nu}(J-\tilde{J}_j)(y^{(j-1)}+c^{(j)}) + \phi(u(j))], \tag{3.2}$$

$$j = M_1, M_2, \ldots .$$

Here, J_H denotes the Jacobian matrix of the right-hand side function corresponding to a coarse grid with grid parameter H. The (rectangular) matrices R_{Hh} and P_{hH} relate the grid functions defined on the coarse grid Ω_H and the grid Ω_h actually used ($h < H$). R_{Hh} (the *restrictor*) transforms a grid function defined on Ω_h into a function defined on Ω_H and P_{hH} (the *prolongator*) vice versa. These operators are assumed to satisfy the relation (cf. [2])

(3.3a) $\|P_{hH}R_{Hh} - I\| = O(H^q)$ as $H \to 0$, $q \geq 1$.

Furthermore, it will be assumed that also

(3.3b) $\|J_H - R_{Hh}J_hP_{hH}\| = O(H^q)$ as $h < H \to 0$.

The coarse grid corrections

(3.1') $c^{(j)} = P_{hH}c_H^{(j)}$, $j = M_1, M_2, \ldots$

require the solution of linear systems for $c_H^{(j)}$ with

(3.4) $A_H = I - b_0\tau^\nu J_H$

as its matrix of coefficients. Let $\tilde{A}_H$ be an approximation to A_H such that $\tilde{A}_H^{-1}$ is easily evaluated and write the linear system for $c_H^{(j)}$ in the form

(3.5) $A_H z = \psi_H^{(j)}$,

then we may define the iteration process (cf.[2])

(3.6a) $z_0 = \tilde{A}_H^{-1}\psi_H^{(j)}$,

(3.6b) $z_\ell = [I - \tilde{A}_H^{-1}A_H]z_{\ell-1} + \tilde{A}_H^{-1}\psi_H^{(j)}$, $\ell = 1,2,\ldots$

We shall assume that this process solves (3.5) with negligible error. Then the iteration error of (3.2) satisfies the equation (cf. (2.2))

$$
\begin{aligned}
(3.7)\qquad [I-b_0\tau^\nu\tilde{J}_j]\varepsilon(j) = b_0\tau^\nu\{&[J-\tilde{J}_j][I-P_{hH}A_H^{-1}R_{Hh}A]\varepsilon(j-1)\\
&- [I + b_0\tau^\nu(J-\tilde{J}_j)P_{hH}A_H^{-1}R_{Hh}][(J\eta - f(t_{n+1},\eta))\\
&\qquad - (Jy^{(u(j))} - f(t_{n+1},y^{(u(j))}))]\}
\end{aligned}
$$

where we have written $A = I - b_0\tau^\nu J$ and where it is assumed that $u(j) = u(j-$ for $j = M_1, M_2, \ldots$.

In a similar way as we derived (2.4) from (2.2) we now derive from (3.7) the estimate

$$\|\varepsilon(j)\| \le |b_0|\tau^\nu \|(I - b_0\tau^\nu \tilde{J}_j)^{-1}\| \{\|J-\tilde{J}_j\| \, \|I-P_{hH}A_H^{-1}R_{Hh}A\| \, \|\varepsilon(j-1)\| + C_j\|I+b_0\tau^\nu(J-\tilde{J}_j)P_{hH}A_H^{-1}R_{Hh}\| \, \|\varepsilon(u(j))\|\}, \tag{3.8}$$

where C_j is defined in the same way as in (2.4). The main difference with the estimate (2.4) is the occurence of the factor $\|I - P_{hH}A_H^{-1}R_{Hh}A\|$ in the error constant of $\varepsilon(j-1)$. In order to see the magnitude of this quantity we substitute A and A_H, and write

$$I - P_{hH}A_H^{-1}R_{Hh}A = (I-b_0\tau^\nu J^*)^{-1}[(I-b_0\tau^\nu J^*) - (I-b_0\tau^\nu J^*)P_{hH}(I-b_0\tau^\nu J_H)^{-1}R_{Hh}(I-b_0\tau^\nu J)],$$

where J^* is a (square) matrix satisfying the relation

$$J^*P_{hH} = P_{hH}J_H. \tag{3.9}$$

It is easily verified that

$$I - P_{hH}A_H^{-1}R_{Hh}A = (I-b_0\tau^\nu J^*)^{-1}[(I-P_{hH}R_{Hh}) - b_0\tau^\nu P_{hH}(J_H-R_{Hh}JP_{hH})R_{Hh}(P_{hH}R_{Hh})^{-1}]. \tag{3.10}$$

Hence, by virtue of (3.3)

$$\|I-P_{hH}A_H^{-1}R_{Hh}A\| = O(H^q+\tau^\nu H^q) \text{ as } \tau, h < H \to 0. \tag{3.11}$$

From (3.8) we now derive for $j = M_1, M_2, \ldots$

$$\|\varepsilon(j)\| \le C\tau^\nu(H^q\|J-\tilde{J}_j\| \, \|\varepsilon(j-1)\| + C_j\|\varepsilon(u(j))\|), \tag{3.12}$$

where C is a uniformly bounded constant as τ and $H \to 0$.

4. NUMERICAL EXPERIMENTS

4.1 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. The grid used to define the coarse grid correction (3.1) has grid parameter $H = 2h$. Thus, $h = 1/10$ results in 81 equations on the fine grid and 16 equations on the coarse grid. For $h = 1/20$ we have 361 and 81 equations, respectively.

The examples were chosen such that the exact solution is available. Therefore, initial and boundary conditions can be prescribed by providing the exact solution.

Our first example is linear and serves to test the effect of the coarse grid correction on the rate of convergence of the iteration scheme:

$$(4.1) \qquad \begin{cases} U_t = \alpha(U_{x_1x_1} + U_{x_2x_2}) - \alpha e^{-t}(4\alpha + x_1^2 + x_2^2), \ \alpha = 1,100, \\ \\ U(x_1,x_2,t) = \alpha e^{-t}(x_1^2 + x_2^2) + 1 \end{cases}$$

Since the exact solution is quadratic in the space variables x_1 and x_2 the space discretization error vanishes so that the time integration aspect can be tested more or less separately from the effects of space discretization.

The second example is defined by [9]

$$(4.2) \qquad \begin{cases} U_t = \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right)U^5 \\ \\ U(x_1,x_2,t) = \left[\frac{4}{5}(2t+x_1+x_2)\right]^{\frac{1}{4}} \end{cases}.$$

This nonlinear problem gives rise to an iteration error $\varepsilon(M)$ where both the error of the inner (Jacobi) iteration and the outer (Newton) iteration are

present (cf. (2.4) and (3.12)). It therefore can be used to demonstrate the effect of the inner and outer iteration processes.

4.2 The numerical scheme.

In this paper the numerical experiments are restricted to parabolic equations, i.e. $\nu = 1$ in (1.1). For the implicit formula (1.2) the fourth order backward differentiation formula (cf. e.g. [6, p.242]) was chosen which results in

$$(4.3) \qquad b_0 = \frac{12}{25}, \quad \zeta_n = \frac{1}{25}\left[48y_n - 36y_{n-1} + 16y_{n-2} - 3y_{n-3}\right]$$

in the iteration process (1.3). This formula was chosen because of its excellent stability properties [1] so that (1.3) is also expected to be stable if the iteration error is sufficiently small in each integration step.

In order to apply {(1.3),(4.3)} four starting values are required which were obtained from the exact solution of the initial-boundary value problems. Furthermore, we put $y^{(pred)} = y_n$, $\tau_n = \tau$ is constant, J was obtained by analytical differentiation, $\tilde{J}_j$ is determined according to (1.8) (for details of the L^*U^*-decomposition used we refer to [12]) and the update function u(j) is defined by

$$(4.4) \qquad u(1) = 0, \; u(j) = \left[\frac{m(j-1)}{M}\right], \qquad j = 2,\ldots,M,$$

where [x] denotes the integer part of x and m is the number of f-evaluations per integration step to be specified in the tables of results.

The scheme {(1.3),(1.8)} was combined with the coarse grid correction (3.2). The values $M_1, M_2, \ldots$ where this coarse grid correction is inserted are given by

$$(4.5) \qquad M_\ell = \ell(p+s+1)-s,$$

where p and s are integers to be specified in the tables of results. From (4.5) it follows that two coarse grid corrections are "separated" by p+s

iterations and the first correction is preceded by p iterations. In the experiments M is always a multiple of p+s, hence the number of coarse grid corrections per integration step is given by

$$r = \frac{M}{p+s} \quad . \tag{4.6}$$

The performance of the coarse grid correction C itself requires the solution of the linear system (3.5) which is solved by (3.6) in μ iterations including the initial iteration (3.6a). The matrix $\tilde{A}_H^{-1}$ in (3.6) is obtained by incomplete LU-decomposition (cf.(1.8)). The prolongator and restrictor operators needed in the coarse grid correction can be compactly formulated by introducing the averaging operators $\mu_-, \mu_|, \mu_+$ an μ_x. When applied to a grid function at a point Q these operators are respectively defined by the average of the values at the two "horizontal", the two "vertical", the four "horizontal" and "vertical" and the four "diagonal" neighbouring points of Q. Furthermore we can divide the grid points into four groups according to figure 4.1. The coarse grid with a parameter H = 2h consists of grid

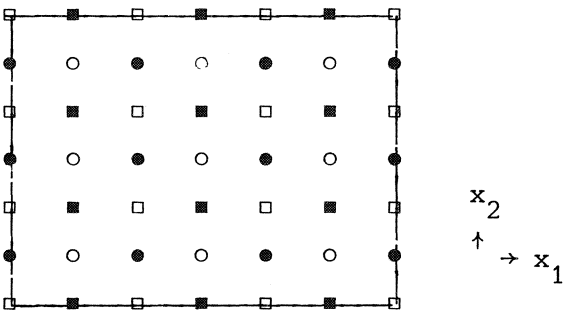


Fig. 4.1 Grid for h = 1/6

points denoted by □. Let v be a grid function defined on the coarse grid, i.e. the points □, then the prolongator is defined by

$$(P_{hH}v)_□ = v;\ (P_{hH}v)_● = \mu_| v;\ (P_{hH}v)_■ = \mu_- v;\ (P_{hH}v)_○ = \mu_x v. \tag{4.7}$$

Let u be a grid function defined on the fine grid with grid parameter h, then

$$(4.8) \qquad (R_{Hh}u)_{\square} = \tfrac{1}{4}u_{\square} + \tfrac{1}{4}(\mu_x u)_{\square} + \tfrac{1}{2}(\mu_+ u)_{\square}.$$

The numerical scheme specified in this section will be called the PCGC (Preconditioning and Coarse Grid Corrections) method.

4.3 Numerical results.

In order to describe exactly what particular PCGC method is used in the tables of results we introduce the following notations (cf. the notation for predictor-corrector methods):

- E evaluation of the function ϕ defined in (1.3)
- I iteration step defined by (1.9)
- C_μ coarse grid correction defined by (3.1) where μ is the number of iterations performed on the coarse grid (cf. section 4.2).

A particular method is now denoted by (cf. section 4.2)

$$(4.9) \qquad \begin{cases} (EI^{M/m})^m & \text{if } r = 0 \\ (E(I^p C_\mu I^s)^{r/m})^m & \text{if } r \geq m \end{cases}$$

Furthermore, we use the notations:

- $A(\tau)$ accuracy in the end point $t = 1$ measured by the minimal number of correct digits, i.e.

$$(4.10) \qquad A(\tau) = -^{10}\log \|y_n - u(t_n)\|_\infty,$$

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

- $\tilde{p}(\tau,2\tau)$ effective order of the scheme $\{(1.3),(1.8)\}$ in the interval $(\tau,2\tau)$ defined by

$$(4.11) \qquad \tilde{p}(\tau,2\tau) = \frac{A(\tau) - A(2\tau)}{^{10}\log 2}.$$

Σ f — total number of f-evaluations given by $(\tau^{-1}-3)m$

Σ matvec — total number of matrix-vector multiplications on Ω_h, which is given by $(\tau^{-1}-3)[m+r(p+s+1)]$ for $r \neq 0$ and by $(\tau^{-1}-3)(m+M)$ for $r = 0$

Σ sol — total number of equations $L^*U^*y = b$ solved on Ω_h, which is given by $(\tau^{-1}-3)(p+s)r$ for $r \neq 0$ and $(\tau^{-1}-3)M$ for $r = 0$.

In the following subsections we show the effect of the coarse grid correction strategy (p,μ,s,r) on the accuracy and the effective order of the scheme for large $\tau\sigma$ values, where σ denotes the spectral radius of the Jacobian matrix J defined in (1.3).

4.3.1 Coarse grid correction strategy

In order to demonstrate the effect of the (p,μ,s,r)-parameters on the accuracy we first choose the linear problem (4.1) which requires only one f-evaluation per step so that the update function is fixed (u(j) = 0). In the tables 4.1 and 4.2 some results are listed showing that inserting coarse

Table 4.1 Results for problem (4.1) with $\alpha = 1$
$m = 1$, $\tau = \frac{1}{4}$ and $h = 1/10$

M	PCGC Method	r	A($\frac{1}{4}$)	Σf	Σ matvec	Σ sol
5	EI^M	0	3.2	1	6	5
10			4.93	1	11	10
4	$EI^2C_6I^2$	1	3.86	1	6	4
2	$E(IC_4I)^r$	1	2.90	1	4	2
4		2	5.26	1	7	4
6		3	4.84	1	10	6

Table 4.2 Results for problem (4.1) with $\alpha = 1$
$m = 1$, $\tau = \frac{1}{4}$ and $h = 1/20$

M	PCGC Method	r	$A(\frac{1}{4})$	Σ f	Σ matvec	Σ sol
5	EI^M	0	1.62	1	6	5
10			2.27	1	11	10
20			3.56	1	21	20
4	$E(I^2C_6I^2)^r$	1	3.09	1	6	4
8		2	5.02	1	11	8
16		4	4.83	1	21	16
2	$E(IC_4I)^r$	1	2.55	1	4	2
4		2	4.46	1	7	4
6		3	4.86	1	10	6
8		4	4.83	1	13	8

Table 4.3 Results for problem (4.1) with $\alpha = 100$
for $m = 1$, $\tau = \frac{1}{4}$ and $h = 1/20$

M	PCGC Method	r	$A(\frac{1}{4})$	Σ f	Σ matvec	Σ sol
2	EI^M	0	-.85	1	3	2
8			-.27	1	9	8
16			.46	1	17	16
2	$E(IC_4I)^r$	1	.40	1	4	2
6		3	3.13	1	10	6
8		4	4.83	1	13	8
4	$E(I^2C_4I^2)^r$	1	.56	1	6	4
12		3	3.71	1	16	12
16		4	4.80	1	21	16
2	$E(IC_8I)^r$	1	.57	1	4	2
6		3	4.30	1	10	6
8		4	4.70	1	13	8

Table 4.4 Values of $A(\frac{1}{4})$ obtained by $(EI^{M/m})^m$ for problem (4.2) with h = 1/20.

M	m=1	m=2	m=3	m=4	m=5
4	1.47				
6	1.68				
8	1.93	1.96			
10	1.96	2.26			
12	1.95	2.44	2.48		
16	⋮	⋮	2.95	3.00	
20	⋮	⋮	⋮	3.48	3.55
24	⋮	⋮	⋮	⋮	3.90
30	⋮	⋮	⋮	⋮	⋮
40	1.95	2.44	2.95	3.48	3.90

grid corrections into the iteration scheme improves the accuracy considerably. They also show that in this rather smooth problem only a few iterations are necessary in the evaluation of the coarse grid corrections.

In table 4.3 the results are given for the highly stiff problem (4.1) with $\alpha = 100$. Without coarse grid corrections (r=0) the convergence is extremely slow, whereas a minimum number of iterations with only a single coarse grid correction is sufficient to obtain some accuracy. Additional experiments have shown that the accuracy gradually increases if the number of coarse grid corrections increases from 1 until 4, and remains constant for larger values of $r(A(\frac{1}{4}) \sim 4.7)$. In this stiff problem the accuracy improves if the number of iterations used in the evaluation of the coarse grid corrections is increased.

Table 4.5 Values of $A(\frac{1}{4})$ obtained by $(E(IC_4I)^{r/m})^m$ for problem (4.2) with h = 1/20.

r=M/2	m=1	m=2	m=3	m=4	m=5
1	1.96				
2	1.95	2.44			
3			2.95		
4				3.48	
5					3.90
6					
8	1.95	2.44	2.95	3.46	3.90

In the tables 4.4 and 4.5 the effect of the number of f-evaluations (m) and coarse grid corrections is illustrated for the non-linear problem (4.2). If no coarse grid corrections are inserted then we see from table 4.4 that roughly M = 4+4m iterations are required in order to reduce the iteration error in the solution of the linear systems to a negligible value. The method $(E(IC_4I)^{r/m})^m$, however, only needs M = 2m iterations to achieve the same result (see table 4.5). Notice that this is also the lowest possible number of iterations because $r \geq m$ and $M = 2r$.

4.3.2 The effective order of the iteration scheme

From experiments with the method of successive corrections {(1.3),(1.5)} reported in [4] it follows that often the order of accuracy is considerably less than the asymptotic order of accuracy, particularly for small values of τ and large values of $\tau\sigma$ where σ denotes the spectral radius of the Jacobian matrix J. Therefore, we are interested in the effective order (4.11) of the scheme {(1.3),(1.8)}.

Table 4.6 Effective order of the method $(EI^3)^m$ for problem (4.2) with h = 1/10.

τ	m=1	m=2	m=3	m=4	m=5
1/5	.6	3.6	4.6	5.9	5.8
1/10	.9	1.3	2.3	-	-
1/20	1.3	1.2	1.6	-	-
1/40					
$\tau \to 0$	p=1	p=2	p=3	p=4	p=4

In table 4.6 the effective orders are given for problem (4.2) together with the asymptotic order p derived from (2.5). The integration steps are performed without coarse grid corrections. Each three iterations the functior f is updated, hence M = 3m. For m > 3 and $\tau \geq 1/20$ the space discretization error becomes dominant in the error $y_n - u(t_n)$ so that (4.11) does not give the order of the time discretization error (indicated by -). The results in this table indicate that the asymptotic order p is not reached. A possible explanation might be the effect of the space discretization error or the still relatively large values of τ. Since we cannot decrease the value of τ (the space discretization error would become dominant) we decrease the valu of h.

Table 4.7 Effective orders of the method $(EIC_4I)^m$ for problem (4.2) with h = 1/20.

τ	m=1	m=2	m=3	m=4	m=5
1/5	1.8	3.7	4.5	5.9	6.7
1/10	2.0	3.3	4.3	4.8	-
1/20	1.0	2.9	4.0	-	-
1/40					
$\tau \to 0$	p=1	p=2	p=3	p=4	p=5

In table 4.7 the results are listed for the method $(EIC_4I)^m$ for h=1/20. Although the asymptotic order is still not shown we see a convergence to the correct values for decreasing τ-values. It is also evident from these results that the introduction of coarse grid corrections increases the effective order considerably.

4.4 Comparison with other integration methods

The PCGC method in $(EIC_4I)^m$ mode has been compared with two other integration methods. The first one is the second order one-step Runge-Kutta-Chebyshev method (RKC method) described in [10]. The second method is also based on the preconditioned linear equation (1.10'), but instead of accelerating Jacobi's method by coarse grid corrections as is done in the PCGC method, the convergence is accelerated by applying Chebyshev iteration (Richardson's method). By virtue of the property that the matrix $I-(L^*U^*)^{-1}R$ has its eigenvalues in the right half-plane (provided that certain mild conditions are satisfied [8]), Manteuffel's analysis of Richardson's method can be applied [7] and the optimal values of the iteration parameters evaluated. The generating method is identical to that of the PCGC method. This method will be called the Preconditioned Richardson method (PR method) and will be denoted by $(EI^q)^m$ where q is the number of iteration steps to solve each linear system to be specified in the tables of results.

Our test examples are again problem (4.1) and (4.2) both with h = 1/20.

In order to compare the three methods the A(τ)-values and the computational effort required are listed in one table. The computational effort is measured by the number N of computational units which are defined differently for each method. For *nonlinear* problems one may choose

$$\begin{aligned} &\text{PCGC(r=m): } f + 4\ \text{matvec} + 2\ \text{sol} + C_4 + \frac{1}{m}[(L^*U^*)_h + (L^*U^*)_H] \\ &\text{RKC: } 10f \\ &\text{PR: } \tfrac{1}{2}[f + (q+1)\ \text{matvec} + \frac{1}{m}(L^*U^*)_h + q\ \text{sol}] \end{aligned} \tag{4.12}$$

Here, L^*U^* denotes the computational effort to perform the incomplete LU-decomposition on Ω_h and Ω_H. For the definition of the other quantities we refer to the preceding subsections.

For *linear* problems one may choose the computational units

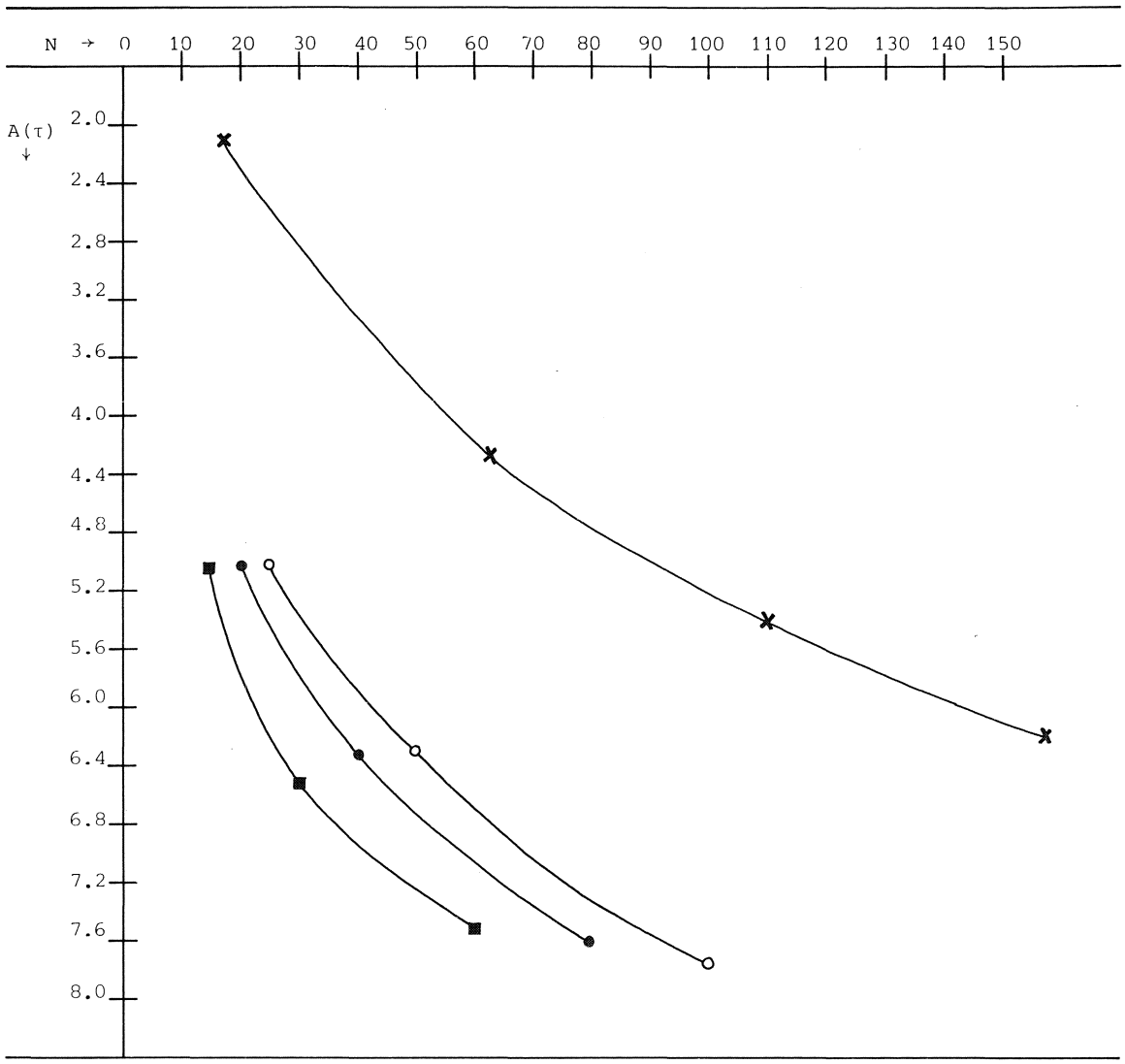
$$
\begin{aligned}
&\text{PCGC: } \frac{1}{r}[\text{f} + \text{matvec}] + C_4 + 3\ \text{matvec} + 2\ \text{sol} \\
&\text{RKC : } 4\text{f} \qquad (4.13) \\
&\text{PR : } \frac{1}{5}[\text{f} + \text{matvec} + \text{q}(\text{matvec} + \text{sol})].
\end{aligned}
$$

Notice that the computational work involved to perform the incomplete LU-decompositions are neglected in these units because for linear problems these calculations are required only once and the decompositions can be used in all integration steps.

In (4.12) and (4.13) not all calculations performed by the various methods are taken into account. In the PCGC method the evaluation of the Jacobian matrices are neglected and are in fact provided in closed form in our experiments; 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 and the evaluation of the Jacobian matrices as well are not taken into account. For a more detailed discussion of the computational units (4.12) and (4.13), and a comparison on the basis of arithmetic operations we refer to [12].

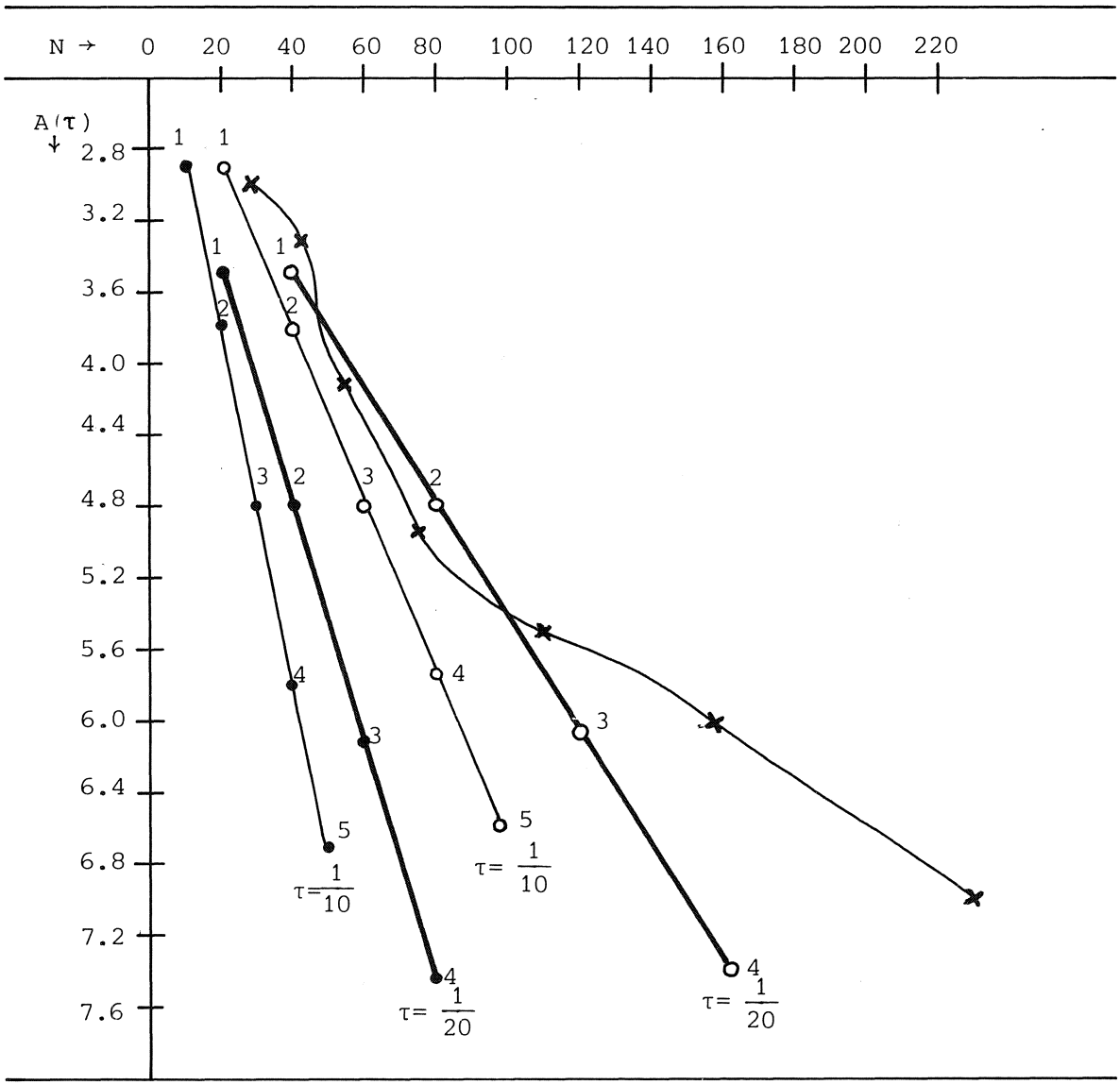
Another important aspect in interpreting the results obtained by the three methods is the *storage requirement*. The RKC method requires only a few vector arrays whereas especially the PCGC method needs considerably more storage.

Table 4.8 $A(\tau)$ and N values obtained
for problem (4.1) with h = 1/20.
× RKC ; ■ $E(IC_4I)^r$ (r=3) ; ● $E(IC_4I)^r$ (r=4) ; ○ EI^q (q=14) ;



In table 4.8 the A(τ) values and the corresponding computational work N (expressed in terms of the units defined in (4.13)) are illustrated. These values were obtained by performing the integration with a number of integration steps (RKC with τ = 1,1/12,1/35 and 1/70, the other methods with τ = 1/5, 1/10 and 1/20).

Table 4.9 A(τ) and N values obtained for problem (4.2) with h = 1/20.
× RKC ; ○ $(EI^q)^m$ (q=9) ; ● $(EIC_4I)^m$;



In table 4.9 the $A(\tau)$ and N values for problem (4.2) are illustrated obtained by the RKC, the PR and the PCGC method. The results of the RKC method correspond to τ = 1,1/2,1/5,1/10,1/20,1/40 and 1/80. For the PR and PCGC methods the integration step τ and the value of m are indicated in the plots.

From the tables 4.8 and 4.9 we conclude that *relative to the units (4.12) and (4.13)* the PCGC method is the most efficient one and the RKC method the most expensive one. However, are the units (4.12) and (4.13) comparable? This is both problem and computer dependent so that we shall not try to answer this question. Moreover, the aspect of storage may be as important as the computational effort which places the RKC at the first place.

Finally, we remark that the PCGC method analysed in this paper should be implemented as a full-multi grid method as described in [2] and [3] if one decides to base a software package on preconditioning and coarse grid corrections. Also a more suitable predictor formula $y^{(pred)}$ might be considered.

REFERENCES

[1] GEAR, C.W., *Numerical initial value problems in ordinary differential equations,* Prentice Hall, Englewood Cliffs, N.J., 1971.

[2] HEMKER, P.W., *Introduction to multi-grid methods,* in MC Syllabus: *Colloquium numerical solution of partial differential equations,* ed. J.G. Verwer, Mathematisch Centrum, Amsterdam, 1980 (to appear in: Nieuw Arch. Wisk).

[3] HEMKER, P.W., *On the structure of an adaptive multi-level algorithm,* BIT 20, pp.289-301, 1980.

[4] HOUWEN, P.J. van der, *Multistep splitting methods of high order for initial value problems,* SIAM J. Numer. Anal., Vol. 17, No. 3, pp. 410-427, 1980.

[5] HOUWEN, P.J., van der, *Multistep splitting methods for nonlinear initial value problems,* in MC Syllabus: *Colloquium numerical solution of partial differential equations,* ed. J.G. Verwer, Mathematisch Centrum, Amsterdam, 1980.

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

[7] MANTEUFFEL, T.A., *The Tchebychev iteration for non-symmetric linear systems*, Num. Math., Vol. 28, pp. 307-327, 1977.

[8] 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. of Comp., Vol. 31, No. 137, pp. 148-162, 1977.

[9] RICHTMYER, R.D. & K.W. MORTON, *Difference methods for initial value problems*, Interscience, New York, 1967.

[10] SOMMEIJER, B.P. & P.J. van der HOUWEN, *On the economization of stabilized Runge-Kutta methods wich applications to parabolic initial value problems*, MC Report NW 74/79, Mathematisch Centrum, Amsterdam, 1979 (to appear in ZAMM, 1981).

[11] STEWART, G.W., *Introduction to matrix computations*, Academic Press, New York, 1973.

[12] VRIES, H.B. de, *The two-level algorithm in the solution of the initial value problem for partial differential equations*, MC Report NW, Mathematisch Centrum, Amsterdam, in preparation.