stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 105/81 APRIL

P.J. VAN DER HOUWEN & B.P. SOMMEIJER

ANALYSIS OF RICHARDSON ITERATION IN MULTIGRID METHODS FOR NONLINEAR PARABOLIC 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: 65M20, 65B99

Analysis of Richardson iteration in multigrid methods for nonlinear para-parabolic differential equations *)

by

P.J. van der Houwen & B.P. Sommeijer

ABSTRACT

The main purpose is the analysis of the behaviour of (nonlinear) Richardson iteration in multigrid methods for solving the implicit relations obtained when an implicit one-step method is applied to the space-discretized form of a (nonlinear) parabolic differential equation. This analysis leads us to an unconventional form of the Richardson method which turns out to be a remarkably efficient predictor formula to start the multigrid method. Numerical experiments are reported obtained by a two-grid algorithm and a comparison is made with (nonlinear) Gauss-Seidel iteration. For large integration steps and for strongly nonlinear problems the Richardson iteration method appears to be superior to Gauss-Seidel iteration.

KEY WORDS & PHRASES: *Numerical analysis, parabolic differential equations, multigrid methods, nonlinear Richardson iteration, Runge-Kutta methods*

---

*) This report will be submitted for publication elsewhere.

1. PRELIMINARIES

1.1. Introduction

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

$$\frac{d\vec{y}}{dt} = \vec{f}(t,\vec{y}), \quad \vec{y}(t_0) = \vec{y}_0 \tag{1.1}$$

originate from the space discretization of an initial-boundary value problem for a (nonlinear) *parabolic* differential equation of the form

$$\frac{\partial U}{\partial t} = G(t,x_1,x_2,U, \frac{\partial U}{\partial x_1}, \frac{\partial^2 U}{\partial x_1^2}, \frac{\partial U}{\partial x_2}, \frac{\partial^2 U}{\partial x_2^2}, \frac{\partial^2 U}{\partial x_1 \partial x_2}). \tag{1.2}$$

Applying some implicit integration formula for ODE's to (1.1) we are faced with the task to solve in each integration step a (nonlinear) system

$$L_h \vec{y}_h = \vec{\Sigma}_h, \tag{1.3a}$$

where L_h is a (nonlinear) operator and $\vec{\Sigma}_h$ is a sum of known terms. Here, h refers to the mesh width of the grid Ω_h used in the space discretization.

In order to accelerate the convergence of the iteration process for solving (1.3a) we will use *defect corrections* obtained by solving an approximate problem on a coarser grid Ω_H. This *coarse grid problem* will be denoted by

$$L_H \vec{y}_H = \vec{\tilde{\Sigma}}_H, \tag{1.3b}$$

where H refers to the grid Ω_H. L_H is the analogue of L_h but now on Ω_H and $\vec{\tilde{\Sigma}}_H$ will be defined later. In [8] the effect of defect corrections was investigated if the fine grid problem (1.3a) and the coarse grid problem (1.3b) are both solved by a *Newton type method*. These methods, however, require the evaluation of the Jacobian matrix of $\vec{f}$ which may be unattractive in view of the large systems presented by (1.1) particularly if two or more space dimensions are involved. In this paper we analyze the behaviour of *nonlinear Richardson iteration processes* for solving (1.3a) and (1.3b). Such methods

are less complicated and require a relatively small amount of storage, but will of course require more $\vec{f}$-evaluations. In view of these storage requirements we will in particular consider Richardson's method for solving one-step implicit integration formulas.

In the following subsection we shortly discuss in general terms the defect correction process as introduced in [14], and in the subsections 1.2 and 1.3 these ideas are applied to our problem (1.3a). In Section 2 the nonlinear Richardson method is discussed and in Section 3 a modification is analysed which turns out to be useful if large eigenvalue intervals of the Jacobian $\partial\vec{f}/\partial\vec{y}$ are involved. Section 4 presents the numerical schemes used in our numerical experiments reported in Section 5. In these experiments we were mainly interested in the behaviour of the iteration method for strongly nonlinear parabolic problems. We did not investigate the effect of using a *multistep* method for generating the problems (1.3a) and (1.3b); in fact, we chose the backward Euler and the trapezoidal rule. We also did not test the recursive use of defect corrections, that is the acceleration of the coarse grid iteration by using defect corrections obtained on a still coarser grid $\Omega_{\tilde{H}}$, etc.

1.2. Defect correction iteration

Consider the problem

$$L\vec{y} = \vec{\Sigma}, \tag{1.4}$$

where L is an arbitrary (nonlinear) operator defined as a finite dimensional vector space and $\vec{\Sigma}$ a given vector. Let $\tilde{L}$ and $\tilde{\vec{\Sigma}}$ be approximations to L and $\vec{\Sigma}$, respectively. Then we may define the (nonlinear) defect correction step [4, 5]

$$\bar{\vec{y}} = \hat{\vec{y}} + \{\tilde{L}^{-1}(\tilde{\vec{\Sigma}} + \vec{\Sigma} - L\hat{\vec{y}}) - \tilde{L}^{-1}\tilde{\vec{\Sigma}}\}, \tag{1.5}$$

where $\hat{\vec{y}}$ is an approximation to the exact solution $\vec{\eta}$ of (1.4). The expression between parenthesis is called the *defect correction*. By using (1.5) repeatedly, we obtain *defect correction iteration*.

In order to see the effect of the defect correction in (1.5) on the iterate $\vec{\hat{y}}$ as an approximation to the solution $\vec{\eta}$ we write

$$\vec{y} = \vec{\hat{y}} + \vec{v} \tag{1.6}$$

and we investigate the vector $\vec{v}$. Evidently, $\vec{v}$ satisfies the equation

$$\tilde{L}(\vec{v} + \tilde{L}^{-1}\vec{\tilde{\Sigma}}) - \tilde{L}\tilde{L}^{-1}\vec{\tilde{\Sigma}} = L\vec{\eta} - L\vec{\hat{y}}. \tag{1.7}$$

Assuming that L and $\tilde{L}$ are differentiable we may write

$$\tilde{L}'\vec{v} = L'(\vec{\eta} - \vec{\hat{y}}) + O(\|\vec{v}\|^2) + O(\|\vec{\eta} - \vec{\hat{y}}\|^2), \tag{1.8}$$

where L' and $\tilde{L}'$ are the Jacobian matrices of L and $\tilde{L}$. Substitution into (1.6) yields

$$\vec{y} - \vec{\eta} = [I - (\tilde{L}')^{-1}L'](\vec{\hat{y}} - \vec{\eta}) + O(\|\vec{\hat{y}} - \vec{\eta}\|^2) \tag{1.9}$$

if $\|\vec{\hat{y}} - \vec{\eta}\|$ is sufficiently small.

Thus, if we can find an approximating operator $\tilde{L}$ such that

$$\|L' - \tilde{L}'\| \ll \|\tilde{L}'\| \tag{1.10}$$

the defect correction process (1.5) is expected to converge very fast in the neighbourhood of the exact solution $\vec{\eta}$.

1.3. Coarse grid corrections

In order to define an efficient defect correction process the application of the operator $\tilde{L}^{-1}$ in (1.5) and the evaluation of $\vec{\tilde{\Sigma}}$ should require considerably less work than the application of the operator L^{-1}. Following BRANDT and HEMKER [1,4,5], we define in the particular case of problem (1.3a)

$$L = L_h, \qquad \vec{\Sigma} = \vec{\Sigma}_h,$$

$$\tilde{L}^{-1} = I_h^H L_H^{-1} I_H^h, \tag{1.11}$$

$$I_H^h \vec{\tilde{\Sigma}} = L_H I_H^h \vec{\hat{y}}_h, \qquad h < H.$$

The corresponding defect correction will be called the *coarse grid correction* on the grid Ω_H. Here, $\vec{\hat{y}}_h$ denotes an approximation to the fine grid solution $\vec{\eta}_h$ (we shall use the indices h and H in order to indicate fine and coarse grid functions). Furthermore, I_h^H and I_H^h are operators (*prolongator* and *restrictor*) which transform grid functions on Ω_H into grid functions on Ω_h and vice versa. Explicit expressions for these operators can be compactly formulated by introducing the averaging operators μ_-, $\mu_|$, μ_+ and 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 divide the grid points into four groups according to Figure 4.1. The coarse grid with grid parameter H = 2h consists of grid points denoted by □. Let $\vec{y}$ be a grid function defined on Ω_H, i.e. the points □, then an often-used prolongation is given by

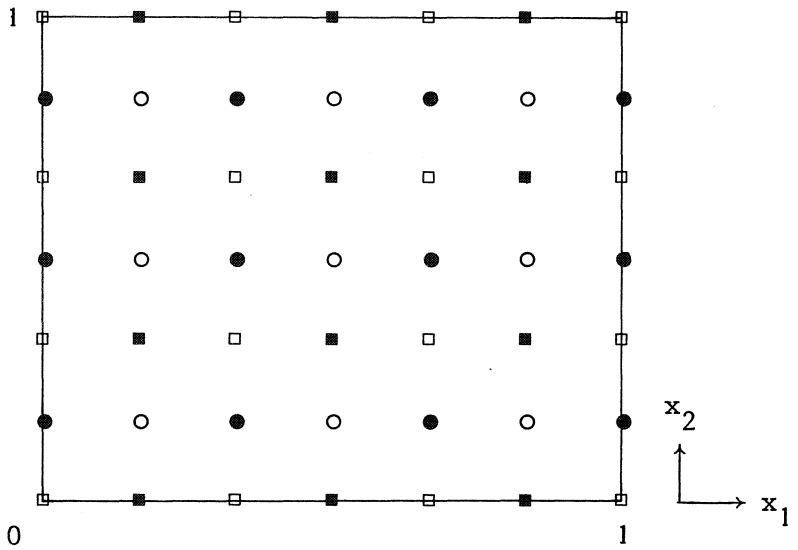

Fig. 4.1. Grid for h = 1/6

$$(1.12)\qquad (I_h^H \vec{y})_{\square} = \vec{y};\quad (I_h^H \vec{y})_{\bullet} = \mu_{|}\vec{y};\quad (I_h^H \vec{y})_{\blacksquare} = \mu_{-}\vec{y};\quad (I_h^H \vec{y})_{\circ} = \mu_x \vec{y}.$$

This operator asks for values of the grid function $\vec{y}$ on the boundary $\partial\Omega_H$ which are not defined because all grid functions occurring in (1.11) are "internal" grid functions. In the application of I_h^H we will use the boundary conditions to define the boundary values of the function $\vec{y}$. Furthermore, let $\vec{y}$ be a grid function defined on Ω_h then an example of a restriction is given by

$$(1.13)\qquad (I_H^h \vec{y})_{\square} = \frac{1}{4}\vec{y}_{\square} + \frac{1}{4}(\mu_x \vec{y})_{\square} + \frac{1}{2}(\mu_{+}\vec{y})_{\square}\ .$$

In order to evaluate the right hand side of (1.5) in the case of (1.11) we have to solve the *coarse grid problem* (1.3b) with

$$(1.14)\qquad \vec{\tilde{\Sigma}}_H = I_H^h \vec{\Sigma}_h + L_H I_H^h \vec{\hat{y}}_h - I_H^h L_h \vec{\hat{y}}_h.$$

Let $\vec{y}_H^{*}$ be the (approximate) solution of this problem then the coarse grid correction step (1.5) reads

$$(1.5')\qquad \vec{\bar{y}}_h = \vec{\hat{y}}_h + I_h^H(\vec{y}_H^{*} - I_H^h \vec{\hat{y}}_h).$$

In the following theorem the effect of the coarse grid correction is given in the case where the operators L_h and L_H are of the form

$$(1.15)\qquad \begin{aligned} L\vec{y} &= \vec{y} - b_0\tau\vec{f}(t,\vec{y}) \\ \vec{f}(t,\vec{y}) &= J(t)\vec{y} + \vec{g}(t) \end{aligned},$$

where $J(t)$ and $\vec{g}(t)$ are an arbitrary matrix and vector, respectively, b_0 is a constant determined by the integration formula used and τ is the integration step.

THEOREM 1. *Let* L_h *and* L_H *be of the form* (1.15) *then* (1.5) - (1.11) *satisfies the relation*

$$\vec{\bar{y}}_h - \vec{\eta}_h = [I - I_h^H I_H^h - b_0\tau I_h^H(I - b_0\tau J_H(t))^{-1}\cdot$$

(1.16)

$$\cdot(J_H(t)I_H^h - I_H^h J_h(t))](\vec{\hat{y}}_h - \vec{\eta}_h). \qquad \square$$

PROOF. Substitution of (1.15) into (1.11) and then working out the defect correction step (1.5) immediately leads to (1.16).

For nonlinear problems relation (1.16) is approximately valid provided that the Jacobian matrix $\partial\vec{f}/\partial\vec{y}$ is slowly varying with $\vec{y}$.

Evidently, the coarse grid function is more effective as $I_h^H I_H^h$ resembles more closely the identity operator and as the operators L_h and L_H are more closely related to each other. This implies that the coarse grid Ω_H should not be that coarse that the operator L_H cannot represent the continuous operator to some extent.

Finally, we consider the difference between the coarse and the fine grid solutions $\vec{\eta}_H$ and $\vec{\eta}_h$ as $H-h \to 0$ and $\tau \to 0$. We shall say that $L_H \to L_h$ as $H-h \to 0$ if $\| L_H I_H^h \vec{y} - I_H^h L_h \vec{y} \| \to 0$ as $H-h \to 0$ for all $\vec{y}$ in the domain of definition of L_h.

THEOREM 1.2. *Let* L_h *and* L_H *be of the form*

$$L = I + \tau A,$$

where A is some nonlinear operator. If A_h, A_H, L_h^{-1} and L_H^{-1} have bounded Lipschitz constants and $L_H \to L_h$, $L_H^{-1} \to L_h^{-1}$ as $H-h \to 0$ then

$$\vec{\eta}_H - I_H^h \vec{\eta}_h = \tau o(1) \qquad \text{as } H-h \to 0. \qquad \square \tag{1.17}$$

PROOF. From the identity

$$(I + \tau A)^{-1} = I - \tau A(I + \tau A)^{-1}$$

which holds for any nonlinear operator A, we derive for the solutions $\vec{\eta}_h$ and $\vec{\eta}_H$ of the problems (1.3a) and (1.3b) that

$$\vec{\eta}_h = \vec{\Sigma}_h - \tau A_h L_h^{-1} \vec{\Sigma}_h$$

$$\vec{\eta}_H = \vec{\tilde{\Sigma}}_H - \tau A_H L_H^{-1} \vec{\tilde{\Sigma}}_H$$

Substitution of (1.14) for $\vec{\tilde{\Sigma}}_H$ yields

$$\vec{\eta}_H - I_H^h \vec{\eta}_h = \tau\{[A_H I_H^h \vec{\hat{y}}_h - I_h^H A_h \vec{\hat{y}}_h]$$

$$+ [I_H^h A_h L_h^{-1} \vec{\Sigma}_h - A_H L_H^{-1} (I_H^h \vec{\Sigma}_h + \tau(A_H I_H^h \vec{\hat{y}}_h - I_h^H A_h \vec{\hat{y}}_h))]\}.$$

It is easily verified that the expression between parentheses is $o(1)$ as $H-h \to 0$ which leads to (1.17). □

1.4. The iteration scheme

Again following HEMKER [4,5] we define a two-level algorithm for solving the fine grid problem (1.3a). This algorithm consists of three basic steps:

(i) Predict an initial approximation $\vec{\hat{y}}_h$ on Ω_h to the fine grid solution $\vec{\eta}_h$;

(ii) Solve the coarse grid problem (1.3b) with $\vec{\hat{y}}_H = I_H^h \vec{\hat{y}}_h$ as initial guess to obtain the approximate solution $\vec{y}_H^*$;

(iii) Solve the fine grid problem (1.3a) with

(1.18) $$\vec{\hat{y}}_h + I_h^H (\vec{y}_H^* - I_H^h \vec{\hat{y}}_h)$$

as initial guess to obtain the approximate solution $\vec{y}_h^{**}$.

If desired the second and third step can be repeated by setting $\vec{\hat{y}}_H = I_H^h \vec{y}_h^{**}$ and $\vec{\hat{y}}_h = \vec{y}_h^*$.

The general idea behind this two-level algorithm is the elimination of the *lower* frequencies from the iteration error in the *coarse grid iteration* (step (ii)) and the elimination of the *higher* frequencies in the *fine grid iteration* (step (iii)). To some extent, this is achieved by any iteration process which eliminates higher frequencies, because the lower frequencies on the fine grid are higher frequencies on the coarse grid. For instance,

(non-linear) Gauss-Seidel type iteration methods could be used. Another class of methods form the Richardson type methods in which the range of frequencies which are to be damped, can easily be adjusted.

In this paper we will in particular analyse Richardson's method for iterating the coarse and fine grid problems (1.3a) and (1.3b)

2. RICHARDSON's METHOD

We restrict our considerations to cases where in the integration step $[t_n,t_{n+1}]$ both the fine grid and the coarse grid problems are of the form (cf. (1.15))

$$\vec{y} - b_0\tau\vec{f}(t_{n+1},\vec{y}) = \vec{\Sigma}, \qquad n = 0,1,2,\ldots, \tag{2.1}$$

with b_0 constant and τ the step length. On the fine grid Ω_h, $\vec{f}(t,\vec{y})$ corresponds to $\vec{f}_h(t,\vec{y})$ and $\vec{\Sigma} = \vec{\Sigma}_h$ follows immediately from the time integrator used. We shall choose the single step implicit formula defined by

$$\vec{\Sigma}_h = \vec{y}_n + (1-b_0)\tau\vec{f}_h(t_n,\vec{y}_n),$$

where $\vec{y}_n$ denotes the numerical solution at $t = t_n$ on Ω_h. On the coarse grid Ω_H, $\vec{f}(t,\vec{y})$ corresponds to $\vec{f}_H(t,\vec{y})$ and $\vec{\Sigma} = \tilde{\vec{\Sigma}}_H$ is defined by (1.14). The exact solution of (2.1) is denoted by $\vec{\eta}$.

The (nonlinear) Richardson method for (2.1) may be defined by (cf. [3])

$$\begin{aligned}
&\vec{y}^{(0)} = \hat{\vec{y}}, \quad \text{some predictor formula for } \vec{\eta},\\
&\vec{y}^{(j+1)} = \mu_j\vec{y}^{(j)} + (1-\mu_j)\vec{y}^{(j-1)} + \lambda_j[\vec{y}^{(j)} - b_0\tau\vec{f}(t_n+\theta_j\tau,y^{(j)}) - \vec{\Sigma}],\\
&\qquad\qquad j = 0,1,\ldots,m-1, \quad \mu_0 = 1,\\
&\vec{y}^{*} = \vec{y}^{(m)}.
\end{aligned} \tag{2.2}$$

In (2.2) the coefficients μ_j and λ_j are given by

$$\mu_j = 2w_0 \frac{T_j(w_0)}{T_{j+1}(w_0)}; \quad \lambda_0 = \frac{w_1}{w_0}; \quad \lambda_j = \frac{w_1}{w_0}\mu_j; \qquad j = 1,2,\ldots,m-1,$$

(2.3)

$$w_0 = \frac{1+\alpha}{1-\alpha} + \frac{2}{b_0\tau\sigma(1-\alpha)}; \quad w_1 = \frac{-2}{b_0\tau\sigma(1-\alpha)}; \quad T_j(w) = \cos[j\arccos w].$$

In the original Richardson method $\vec{f}$ is linear in $\vec{y}$ and all parameters θ_j equal 1; in addition, α is defined

$$\alpha = \frac{\delta}{\sigma}, \tag{2.4}$$

where $[-\sigma,-\delta]$ is the (negative) eigenvalue interval of the Jacobian matrix $f_y = \partial\vec{f}/\partial\vec{y}$ of the right hand side function. In that case the iteration error is given by

$$\vec{y}^{(j)} - \vec{\eta} = P_j(\tau f_y)[\vec{\hat{y}} - \vec{\eta}], \tag{2.5}$$

where $P_j(z)$ is the polynomial of degree j in z with a minimal maximum norm in the interval $[-\tau\sigma,-\tau\alpha\sigma]$ and such that $P_j(1/b_0) = 1$. This polynomial is given by

$$P_j(z) = \frac{T_j(w_0+w_1-w_1b_0z)}{T_j(w_0)}. \tag{2.6}$$

Thus, Richardson's method with $\alpha = \delta/\sigma$ achieves an optimal damping of the eigenvector components in the initial error $\vec{\hat{y}} - \vec{\eta}$. The damping factor is given by

$$\underset{-\tau\sigma\le z\le-\tau\delta}{\text{maximum}} |P_j(z)| = T_j^{-1}(w_0) = (\cosh[j\,\text{arccosh}\, w_0])^{-1}. \tag{2.7}$$

For $w_0 \le 1.05$ one may use the approximation

$$T_j(w_0) = \cosh[j\sqrt{2(w_0-1)}](1+\varepsilon_j), \quad |\varepsilon_j| < .0015j \tag{2.7'}$$

to get an impression of the damping factor after j iterations.

If $\vec{f}$ is *nonlinear*, Richardson's method can formally be defined by (2.2) - (2.4) if δ and σ are assumed to be the smallest and largest eigenvalue

(in absolute value) of the Jacobian f_y at $(t_n,\vec{y}_n)$. The iteration error is then approximately given by (2.5) only if second and higher order powers of the iteration error are neglected (cf. Section 3.1).

An unattractive aspect of the Richardson method is the evaluation of the eigenvalues σ and δ. The value of σ (spectral radius) may be obtained by a power method as described in [10,15], but the value of δ is more difficult. In this paper we do not suppose that δ is available and consider the scheme {(2.2) - (2.3)} with α in the interval

$$-(b_0\tau\sigma)^{-1} < \alpha < 1 \tag{2.8}$$

in order to have a damping factor less than 1.

3. ACCURACY IN THE NONLINEAR RICHARDSON METHOD

In this section we consider the accuracy of Richardson's method for solving problems of the special form (2.1) originating from parabolic differential equations.

3.1. The iteration error

In the following the iteration error is denoted by

$$\vec{\varepsilon}_j = \vec{y}^{(j)} - \vec{\eta} \tag{3.1}$$

and $T(z)$ denotes the operator defined by

$$T(z)\vec{\varepsilon}_j = \vec{\varepsilon}_{j+1} - (\lambda_j+\mu_j-\lambda_j b_0 z)\vec{\varepsilon}_j - (1-\mu_j)\vec{\varepsilon}_{j-1}. \tag{3.2}$$

Furthermore, all derivatives of $\vec{f}$ are assumed to be evaluated at $(t_{n+1},\vec{\eta})$ unless otherwise stated.

Let us write $\vec{\varepsilon}_j$ as the sum of the error (2.5) obtained for the *original* Richardson method plus some perturbation $\vec{\delta}_j$, i.e.

$$\vec{\varepsilon}_j = P_j(\tau f_y)\vec{\varepsilon}_0 + \vec{\delta}_j, \qquad \vec{\delta}_0 = \vec{0}, \tag{3.3}$$

where f_y denotes the matrix $\partial\vec{f}/\partial\vec{y}$. Substitution into (3.2) yields

$$T(0)\vec{\delta}_j = -T(0)P_j(\tau f_y)\vec{\varepsilon}_0 + \lambda_j b_0 \tau\{[\vec{f}(t_{n+1},\vec{\eta}) - \vec{f}(t_n+\theta_j\tau,\vec{\eta})] + [\vec{f}(t_n+\theta_j\tau,\vec{\eta}) - \vec{f}(t_n+\theta_j\tau,\vec{y}^{(j)})]\}.$$

By linearizing $\vec{f}$ with respect to $\vec{y}$ and observing that $P_j(z)$ satisfies

$$T(z)P_j(z) = 0 \tag{3.4}$$

we obtain

LEMMA 3.1. *The iteration error of* (2.2) - (2.3) *is of the form* (3.3) *where*

$$T(\tau f_y)\vec{\delta}_j = \lambda_j b_0 \tau\{[\vec{f}(t_{n+1},\vec{\eta}) - \vec{f}(t_n+\theta_j\tau,\vec{\eta})] + [f_y(t_n,\vec{\eta}) - f_y(t_n+\theta_j\tau,\vec{\eta})]\vec{\varepsilon}_j + O(\|\varepsilon_j\|^2)\}. \tag{3.5}$$

□

If $\vec{f}(t,\vec{y})$ is independent of t and linear in $\vec{y}$ we see from (3.5) that $\vec{\delta}_j = \vec{0}$ is a solution satisfying the initial condition $\vec{\delta}_0 = \vec{0}$. Then, (3.3) reduces to the formula (2.5). For nonlinear functions $\vec{f}$ nonzero perturbations $\vec{\delta}_j$ may be obtained. In this paper we consider the case where second order expressions in $\vec{\varepsilon}_j$ can be neglected, for example if $\vec{f}(t,\vec{y})$ is linear in $\vec{y}$, i.e.

$$\vec{f}(t,\vec{y}) = J(t)\vec{y} + \vec{g}(t) \tag{3.6}$$

with J(t) an arbitrary matrix and $\vec{g}(t)$ an arbitrary function. Notice that $\vec{\delta}_j$ always vanishes if $\theta_j = 1$ for all j.

THEOREM 3.1. *For the function* (3.6) *the iteration error of Richardson's method* (2.2) *is of the form*

$$\vec{\varepsilon}_j = P_j(\tau f_y)\vec{\varepsilon}_0 + \vec{\delta}_j,$$
$$\vec{\delta}_j = \tau^2[Q_j(\tau f_y)\vec{f}_t + \tau R_j(\tau f_y)\vec{f}_{tt} + S_j(\tau f_y)f_{ty}\vec{\varepsilon}_0 + \tau W_j(\tau f_y)f_{tty}\vec{\varepsilon}_0 + O(\tau^2)] \tag{3.7}$$

where Q_j, R_j, S_j *and* W_j *are polynomials in the matrix* τf_y *defined by*

$$
\begin{aligned}
&Q_0(z) = 0, \quad T(z)Q_j(z) = \lambda_j b_0(1-\theta_j) \\
&R_0(z) = 0, \quad T(z)R_j(z) = \tfrac{1}{2}\lambda_j b_0(1-\theta_j)^2 \\
&S_0(z) = 0, \quad T(z)S_j(z) = \lambda_j b_0(1-\theta_j)P_j(z) \\
&W_0(z) = 0, \quad T(z)W_j(z) = \tfrac{1}{2}\lambda_j b_0(1-\theta_j)^2 P_j(z).
\end{aligned}
\tag{3.8}
$$

PROOF. Substitution of the perturbation $\vec{\delta}_j$ determined by (3.7) into (3.5) shows that (3.5) is satisfied if Q_j, R_j, S_j and W_j satisfy (3.8).

In the next subsection this theorem will be applied for deriving the approximation error of the integration method.

3.2. The approximation error

In the preceding subsection we concentrated on the Richardson method as an iteration process for solving the system (2.1). Next we investigate with what accuracy the solution of the initial-boundary value problem satisfies the numerical scheme. In other words, with what error does the numerical scheme (2.2) - (2.3) approximate the partial differential equation (1.2) for h and $\tau \to 0$. In particular, we analyse the time-discretization part and we will assume that the exact solution $\vec{y}(t)$ of (1.1) and the exact solution U of the initial-boundary value problem (1.2) satisfy the condition

$$
\vec{Y}(t) - \vec{y}(t) \to 0 \qquad \text{as } H \text{ or } h \to 0, \tag{3.9}
$$

where $\vec{Y}(t)$ denotes the function $U(t,x_1,x_2)$ when restricted to the (internal) grid Ω. In this connection we emphasize that the right hand side function $\vec{f}(t,\vec{v})$ need not remain bounded for an arbitrary grid function $\vec{v}$ if the grid is refined. Only for grid functions $\vec{v}$ which match the boundary conditions, such as $\vec{Y}$ and by virtue of (3.9) the solution $\vec{y}$ of (1.1), the function $\vec{f}$ will converge [2,13]. This observation turns out to be crucial in the following analysis of the Richardson method (see below).

Let us write the Richardson method in the form (using (3.3))

$$\frac{\vec{y}^{*}-\vec{\eta}}{\tau} - P_m(\tau f_y)\frac{\hat{\vec{y}}-\vec{\eta}}{\tau} - \frac{\vec{\delta}_m}{\tau} = 0. \tag{3.10}$$

We want to relate this discrete equation to the differential equation (1.1). Therefore, we substitute for the numerical solution at t_n, $n = 0,1,\ldots$ the exact solution $\vec{y}(t)$ at t_n (the *localizing assumption* (cf. [9, p. 27])). The residual left will be called the *approximation error* and is denoted by $\vec{A}_n$. On the fine grid the numerical solution at t_n and t_{n+1} are defined by $\vec{y}_n$ and $\vec{y}_h^{*}$ and on the coarse grid by $I_H^h \vec{y}_n$ and $\vec{y}_H^{*}$, hence

$$\vec{A}_n = \frac{\vec{y}(t_{n+1})-\vec{\eta}}{\tau} - P_m(\tau f_y)\frac{\hat{\vec{y}}-\vec{\eta}}{\tau} - \frac{\vec{\delta}_m}{\tau}\,. \tag{3.10'}$$

The error $\vec{A}_n$ describes the relation between the fine or coarse grid processes and the continuous problem (1.1).

Let us consider this error for the test model (3.6). We assume that $\vec{\eta}_h$ and $\hat{\vec{y}}_h$ have the expansions

$$\begin{aligned} \vec{\eta}_h &= \vec{y}_h(t_n) + \tau\dot{\vec{y}}_h(t_n) + \beta_2\tau^2\ddot{\vec{y}}_h(t_n) + O(\tau^3) \\ \hat{\vec{y}}_h &= \vec{y}_h(t_n) + \hat{\beta}_1\tau\dot{\vec{y}}_h(t_n) + \hat{\beta}_2\tau^2\ddot{\vec{y}}_h(t_n) + O(\tau^3) \end{aligned} \qquad \text{as } \tau \to 0. \tag{3.11a}$$

By assuming that the conditions of Theorem 1.2 are satisfied we find from (1.17) that the coarse grid solution has the expansion

$$\begin{aligned} \vec{\eta}_H = \vec{y}_H(t_n) &+ \tau\dot{\vec{y}}_H(t_n) + \beta_2\tau^2\ddot{\vec{y}}_H(t_n) \\ &+ \left(1+\tau\frac{d}{dt}+\beta_2\tau^2\frac{d^2}{dt^2}\right)\left(I_H^h\vec{y}_h(t)-\vec{y}_H(t)\right)\Big|_{t=t_n} + \tau o(1) + O(\tau^3) \end{aligned}$$

as $H-h \to 0$ and $\tau \to 0$. A similar expansion is obtained for $\hat{\vec{y}}_H = I_H^h\hat{\vec{y}}_h$. By virtue of our assumption (3.9) we find

$$\begin{aligned} \vec{\eta}_H &= \vec{y}_H(t_n) + \tau\dot{\vec{y}}_H(t_n) + \beta_2\tau^2\ddot{\vec{y}}_H(t_n) + \vec{\rho}_H + O(\tau^3) \\ \hat{\vec{y}}_H &= \vec{y}_H(t_n) + \hat{\beta}_1\tau\dot{\vec{y}}_H(t_n) + \hat{\beta}_2\tau^2\ddot{\vec{y}}_H(t_n) + \hat{\vec{\rho}}_H + O(\tau^3) \end{aligned} \qquad \text{as } \tau \to 0, \tag{3.11b}$$

where

$$\vec{\rho}_H = I_H^h \vec{y}_h(t_n) - \vec{y}_H(t_n) + \tau o(1), \quad \vec{\hat{\rho}}_H = \vec{\rho}_H + \tau o(1) \quad \text{as } H-h \to 0.$$

Note that $\vec{\rho}_H$ and $\vec{\hat{\rho}}_H$ both vanish as $h < H \to 0$ but they do not if $\tau \to 0$ and H is kept fixed.

Substitution of (3.11) and (3.10') and using Theorem 3.1 yields the following lemma:

LEMMA 3.2. *Under the assumptions* (3.11) *the approximation error for the test equation* (3.6) *is given by*

$$\begin{aligned} \vec{A}_n &= (1-\hat{\beta}_1)P_m(0)\dot{\vec{y}}(t_n) - \tau Q_m(0)\vec{f}_t(t_n,\vec{y}(t_n)) \\ &+ \tau[(\tfrac{1}{2}-\beta_2+(\beta_2-\hat{\beta}_2)P_m(0))\ddot{\vec{y}}(t_n) - (\hat{\beta}_1-1)P_m'(0)J(t_n)\dot{\vec{y}}(t_n)] \\ &+ \vec{C}_n + O(\tau^2) \qquad \text{as } \tau \to 0, \end{aligned} \tag{3.12}$$

where $\vec{C}_n$ *vanishes on the fine grid and is given by*

$$\begin{aligned} \vec{C}_n &= \tau^{-1}[-\vec{\rho}_H - (P_m(0)+\tau P_m'(0)J(t_n)+\tau^2 S_m(0)J'(t_n))(\vec{\hat{\rho}}_H - \vec{\rho}_H)] \\ &= \tau^{-1}[\vec{y}_H(t_n) - I_H^h \vec{y}_h(t_n)] + o(1) \qquad \text{as } H-h \to 0 \end{aligned}$$

on the coarse grid. □

The (coarse grid vector) $\vec{C}_n$ may also be neglected on the coarse grid if H is sufficiently small with respect to τ. Then, the most important part of the approximation error consists of the other terms in (3.12) which heavily depend on the choice of the initial approximation $\vec{\hat{y}}$. We shall consider the first and subsequent defect corrections in more detail.

In the *first* defect correction step the initial approximation $\vec{\hat{y}}_h$ has to be provided. If one-step formulas are used for generating the problems (1.3), then the most plausible choice for starting the two-level algorithm is defined by

$$(3.13)\qquad \hat{\vec{y}}_h = \vec{y}_n, \quad \hat{\vec{y}}_H = I_H^h \vec{y}_n.$$

With this choice we have $\hat{\beta}_1 = 0$ and the first term in (3.12) reads $P_m(0)\dot{\vec{y}}(t_n)$. This term is $O(\tau^m)$ as $\tau \to 0$, but due to the large order constant if the spectral radius σ of $J(t)$ is large (this happens if h or H are small), it is not necessarily small. To see this we consider the i-th derivative $P_m^{(i)}(0)$ of $P_m(z)$ at $z = 0$ for large values of $\tau\sigma$. It is easily derived from the definition of Chebyshev polynomials that $(-1/b_0\tau\sigma < \alpha < 1)$

$$(3.14)\qquad P_j^{(i)}(0) = \left[\frac{2}{(1-\alpha)\tau\sigma}\right]^i \frac{T_j^{(i)}\left(\frac{1+\alpha}{1-\alpha}\right)}{T_j\left(\frac{1+\alpha}{1-\alpha} + \frac{2}{(1-\alpha)b_0\tau\sigma}\right)} = \frac{\text{const.}}{(\tau\sigma)^i} \qquad \text{as } \tau\sigma \to \infty.$$

Thus, for small values of τ and h or H, the first term in (3.12) may still be large unless we choose α such that $P_m(0) = 0$. Identifying $(1+\alpha)/(1-\alpha)$ with the largest zero of $T_m(z)$ yields

$$(3.15a)\qquad \alpha = \alpha_0 = \frac{\cos\frac{\pi}{2m} - 1}{\cos\frac{\pi}{2m} + 1}.$$

However, even for $P_m(0) = 0$ the approximation error may be large due to the large norm of the Jacobian $J(t_n)$ and the vector $\vec{f}_t$ as h or $H \to 0$. As an example, we consider the one-dimensional diffusion equation

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \qquad U(t,0) = g_0(t), \qquad U(t,1) = g_1(t).$$

Semi-discretization on a uniform grid $\Omega_h = \{ih\}_{i=1}^{1/h-1}$ leads to a system of the form (1.1) with right hand side function of the form (3.6) where

$$J(t) = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}.$$

Evidently, the vector $J\dot{\vec{y}}(t_n)$ occurring in (3.12) has an increasing norm as $h \to 0$ unless $\vec{y}(t_n)$ has zero-boundary values which means that the *boundary conditions are time-independent*. Suppose, however that the polynomial $Q_m(z)$ is such that $Q_m(0) = -P_m'(0)$, then by observing that for (3.6)

$$\vec{f}_t(t,\vec{y}(t)) = \ddot{\vec{y}}(t) - J(t)\dot{\vec{y}}(t)$$

the approximation error reduces to

$$(3.12') \qquad \vec{A}_n = \tau[\tfrac{1}{2} - \beta_2 + P_m'(0)]\dddot{\vec{y}}(t_n) + \vec{C}_n + O(\tau^2) \qquad \text{as } \tau \to 0.$$

THEOREM 3.2. *Let* α, $\vec{f}$ *and* $\hat{\vec{y}}$ *be defined by* (3.15a), (3.6) *and* (3.13), *respectively. Then the approximation error is given by* (3.12') *if*

$$(3.15b) \qquad \theta_j = 1 - P_j(0), \qquad j = 0,1,\ldots,m-1. \qquad \square$$

PROOF. By Theorem 3.1 and by virtue of (3.15b) we find that $Q_j(0)$ satisfies the relation

$$Q_0(0) = 0, \quad T(0)Q_j(0) = \lambda_j b_0(1-\theta_j) = \lambda_j b_0 P_j(0).$$

Since $P_j'(0)$ satisfies the relation

$$P_0'(0) = 0, \quad T(0)P_j'(0) = -\lambda_j b_0 P_j(0)$$

it follows that $Q_j(0) = -P_j'(0)$ which leads to (3.12'). $\square$

Thus, if (3.13) is chosen to start the integration of the interval $[t_n, t_{n+1}]$ then Theorem 3.2 tells us how to tune Richardson's method such that the appproximation error is relatively small. An alternative is of course the application of initial approximations which are first order in time. In view of the large spectral radius σ associated with semi-discrete parabolic equations the formulas generating these initial approximations should be sufficiently stable. For instance, one may use the stabilized Runge-Kutta methods as presented in [6]. (Methods such as forward Euler are not suitable unless

the integration step is unrealisticly small.) Let us assume that $\vec{\hat{y}}$ does have first order accuracy in time (in the second and subsequent defect correction steps this is always true), then by putting $\hat{\beta}_1 = 1$ in (3.12) we find

$$\vec{A}_n = \tau[(\tfrac{1}{2} - \beta_2 + (\beta_2 - \hat{\beta}_2)P_m(0))\ddot{\vec{y}}(t_n) - Q_m(0)\vec{f}_t(t_n, \vec{y}(t_n))] + \vec{C}_n + O(\tau^2) \quad (3.12'')$$

as $\tau \to 0$.

By the same argument as above we conclude that for h or $H \to 0$ the approximation error will be large unless $\vec{f}_t$ vanishes (autonomous systems) or $Q_m(0) = 0$. The latter will be the case if $\theta_j = 1$ for all j (see (3.8)). We observe that the approximation error reduces to $\vec{C}_n + O(\tau^2)$ if we choose the starting formula such that $\hat{\beta}_2$ satisfies the relation

$$\tfrac{1}{2} - \beta_2 + (\beta_2 - \hat{\beta}_2)P_m(0) = 0.$$

Summarizing the results of this section, we have found that the Richardson iteration method (2.2) - (2.3) should satisfy the conditions (3.15a) and (3.15b) if the initial approximation $\vec{\hat{y}}$ is zero order in time and the condition $\theta_j = 1$ if $\vec{\hat{y}}$ is of higher order.

Finally, it should be remarked that the Richardson iteration method (2.2) - (2.3) satisfying (3.15a) and (3.15b) can be interpreted as an *first order consistent, m-stage Runge-Kutta method with τ-dependent coefficients*. It presents an integration method in its own right and might also be of interest as an explicit, stabilized time integrator for parabolic differential equations when only one grid is used. Results of this type of "Runge-Kutta-Richardson methods" are published in the near future.

3.3. Applications

By a few numerical examples we will illustrate the effect of the conditions (3.15) on the behaviour of Richardson's method (2.2) - (2.3).

3.3.1. A linear test problem

In order to test the linear analysis given in the preceding subsection we integrate the following linear equation

$$\frac{\partial U}{\partial t} = (2 + \sin(\omega x_1 x_2))\Delta U$$

$$+ e^{-t}\sin(\omega x_1 x_2)[\omega^2(2+\sin(\omega x_1 x_2))(x_1^2 + x_2^2) - 1], \qquad \omega = 10 \tag{3.16a}$$

with exact solution

$$U(t,\vec{x}) = 1 + e^{-t}\sin(\omega x_1 x_2). \tag{3.16b}$$

Dirichlet boundary conditions along the unit square and initial values at $t = 0$ are determined by (3.16b). This problem is semi-discretized on Ω_h with square meshes of width h by using the standard five point molecule. The resulting system of O.D.E.'s has a Jacobian with negative eigenvalues in the interval $[-\sigma,-\delta]$ where

$$\sigma \cong \frac{24}{h^2}, \quad \delta \cong 2\pi^2. \tag{3.16c}$$

Choosing a simple one-step, one-point integration formula the problem (2.1) on the grid Ω_h assumes the form

$$\vec{y} - b_0\tau\vec{f}(t_{n+1},\vec{y}) = \vec{y}_n + (1-b_0)\tau\vec{f}(t_n,\vec{y}_n). \tag{3.17}$$

To this linear system of equations we applied the Richardson method (2.2) - (2.3) for a few choices of the parameter α and the parameters θ_j. To get an impression of the behaviour of Richardson iteration as a process on its own we did not yet combine the fine grid iteration with coarse grid corrections.

In order to interpret the number of iterations used in the iteration process it is convenient to express m in terms of the parameter α and the damping parameter

$$D = T_m^{-1}(w_0).$$

From the definition of Chebyshev polynomials it follows that

$$(3.18) \qquad m = \frac{\operatorname{arccosh} D^{-1}}{\ell n(1+\alpha+2\varepsilon+2\sqrt{(1+\varepsilon)(\alpha+\varepsilon)})-\ell n(1-\alpha)}, \quad \varepsilon = \frac{1}{b_0\tau\sigma}.$$

From this expression we derive

$$(3.18') \qquad m \cong \frac{\operatorname{arccosh} D^{-1}}{\ell n(\frac{1+\sqrt{\alpha}}{1-\sqrt{\alpha}})} \qquad \text{if } |\alpha b_0\tau\sigma| >> 1.$$

Hence low damping ($D \simeq 1$) or larger values of α result in small m-values. In the special case where $\alpha = \alpha_0$ given by (3.15a), we find

$$(3.18'') \qquad m \cong \frac{1}{4}\sqrt{b_0\tau\sigma[4\operatorname{arccosh}^2(D^{-1})+\pi^2]} \qquad \text{if} \quad \begin{array}{l} \text{either } b_0\tau\sigma \geq 40 \\ \text{or } D^{-1} \geq \cosh(\frac{1}{2}\pi\sqrt{\frac{40-b_0\tau\sigma}{b_0\tau\sigma}}). \end{array}$$

Thus, in this case we see that at least $\pi\sqrt{b_0\tau\sigma}/4$ iterations are required (choose D = 1). If higher damping is desired (D < 1) more iterations are required. In view of the large $\tau\sigma$-values usually involved in actual problems, the "Runge-Kutta-Richardson" method will require a large number of right hand side evaluations. Nevertheless, this method turns out to be more efficient than the original Richardson method. This is illustrated in the tables below where we applied the Richardson iteration method in several modes:

$$(3.19) \qquad \begin{array}{l} R(D,\alpha) \text{ mode defined by (2.2), (2.3) with } \theta_j = 1,\ j = 0,1,\ldots,m-1 \\ RKR(D) \text{ mode defined by (2.2), (2.3) and (3.15a), (3.15b)} \end{array}.$$

Both modes were applied with starting vectors defined by (3.13). In these tables the number of correct digits at t = 1, defined by

$$(3.20) \qquad A = \min_{\Omega_h}\left[-^{10}\log |\text{exact solution of (1.2)} - \text{numerical solution}|\right],$$

is given for a sequence of τ-values and for Richardson's method both in *R* and *RKR* mode. In the *RKR* mode the number of iterations per integration step was chosen equal to $\sqrt{b_0\tau\sigma/1.55}$ (cf. (3.18")). However, requiring m to be an integer we obtain marginal damping. (In these examples D turned out to be at most 0.94.) In the *R* mode, D was chosen in such a way that the total number of iterations equals that of the *RKR* mode for each step size τ. The total number of $\vec{f}$-evaluations is denoted by Σm.

It is obvious from the Tables 3.1, 3.2 and 3.3 that the $R(\leq .94, \alpha_0)$ process is of no value indicating how important the term $J(t_n)\dot{\vec{y}}(t_n)$ is in the approximation error (3.12). Of course, by increasing the damping the results will be improved, however, at the cost of many $\vec{f}$-evaluations. It is also evident that the *RKR* mode becomes superior to the *R* mode if h decreases, i.e. if $\tau\sigma$ increases.

TABLE 3.1

A-values at t=1 obtained by the iterated Euler rule (b_0=1) with $\hat{\vec{y}} = \vec{y}_n$ and h = 1/10

τ	$\tau\sigma$	Σm	$R(D,\delta/\sigma)$	D	$R(\leq .94,\alpha_0)$	$RKR(\leq .94)$
1	2400	40	1.6	.00125	.0	1.1
1/5	480	90	1.6	.053	.8	1.6
1/10	240	130	1.7	.125	.9	1.7

TABLE 3.2

A-values at t=1 obtained by the iterated Euler rule (b_0=1) with $\hat{\vec{y}} = \vec{y}_n$ and h = 1/20

τ	$\tau\sigma$	Σm	$R(D,\delta/\sigma)$	D	$R(\leq .94,\alpha_0)$	$RKR(\leq .94)$
1	9600	79	2.1	.00125	.0	1.8
1/5	1920	180	2.1	.053	.6	2.2
1/10	960	250	2.3	.125	.7	2.4

TABLE 3.3.

A-values at t=1 obtained by the iterated Euler rule (b_0=1) with $\hat{\vec{y}} = \vec{y}_n$ and h = 1/40

τ	$\tau\sigma$	Σm	$R(D,\delta/\sigma)$	D	$R(\leq .94,\alpha_0)$	$RKR(\leq .94)$
1	38400	158	2.3	.00125	.0	2.4
1/5	7680	355	2.1	.053	.6	3.1
1/10	3840	500	2.3	.125	.6	3.1

3.3.2. A strongly nonlinear problem

In this section we show that the linear analysis given in the preceding subsections is also indicative of the behaviour of Richardson's method in strongly nonlinear problems. As a test model we choose the equation [12]

$$\frac{\partial U}{\partial t} = \Delta(U^5). \tag{3.21a}$$

Just as in problem (3.16) we prescribe Dirichlet boundary conditions on the unit square $\Omega = \{\vec{x} \mid 0 \leq x_1,\ x_1 \leq 1\}$ and initial conditions for t = 0 by prescribing the exact solution

$$U(t,\vec{x}) = [\frac{4}{5}\ (2t+x_1+x_2)]^{1/4}. \tag{3.21b}$$

The semi-discretization process yields a system of nonlinear O.D.E.'s of which the Jacobian matrix has its eigenvalues in the interval $[-\sigma(t),-\delta(t)]$ where

$$\sigma(t) \cong \frac{64}{h^2}\ (1+t), \qquad \delta(t) \cong 16\pi^2(t+h). \tag{3.21c}$$

The results listed in the Tables 3.4 and 3.5 reveal that the same behaviour is exhibited as for the linear problem (3.16) but even more pronunciated. In Table 3.6 results are given for the trapezoidal rule; here we see that the accuracy of the *RKR* process decreases if τ decreases, whereas the *R* process becomes more accurate. The reason for this behaviour is the

instability developed by the *RKR* mode (being in fact an explicit Runge-Kutta method) because the stability condition is violated (in the Euler case the stability condition is just satisfied). The stability analysis of *RKR* methods as integration methods on their own will be reported in a next paper where it is shown that stability for the iterated trapezoidal rule requires a considerable larger number of iterations. Since in this paper, we use the *RKR* method as part of a multigrid algorithm and not as a special type of Runge-Kutta method, we are not concerned with the stability of the *RKR* method but only with the stability of the integration formula generating the problem (2.1).

TABLE 3.4

A-values at t=1 obtained by the iterated Euler rule (b_0=1) for h = 1/10 and $\hat{\vec{y}} = \vec{y}_n$

τ	$\tau\sigma$	Σm	$R(D,\delta/\sigma)$	D	$R(\leq .94,\alpha_0)$	$RKR(\leq .94)$
1	12800	91	$-\infty$	.05	$-\infty$	3.5
1/5	2560	184	2.4	.039	1.4	4.6
1/10	1280	258	2.9	.073	1.7	5.1

TABLE 3.5

A-values at t=1 obtained by the iterated Euler rule (b_0=1) for h = 1/20 and $\hat{\vec{y}} = \vec{y}_n$

τ	$\tau\sigma$	Σm	$R(D,\delta/\sigma)$	D	$R(\leq .94,\alpha_0)$	$RKR(\leq .94)$
1	51200	182	$-\infty$	.050	$-\infty$	3.5
1/5	10240	365	2.3	.038	$-\infty$	4.5
1/10	5120	509	2.8	.074	$-\infty$	5.1

TABLE 3.6

A-values at t=1 obtained by the iterated trapezoidal rule ($b_0 = \frac{1}{2}$) for h = 1/10 and $\vec{\hat{y}} = \vec{y}_n$

τ	$\tau\sigma$	Σm	$R(D,\delta/\sigma)$	D	$R(\leq .94,\alpha_0)$	$RKR(\leq .94)$
1	12800	65	$-\infty$	.050	$-\infty$	3.4
1/5	2560	131	2.4	.038	$-\infty$	2.9
1/10	1280	184	2.6	.073	$-\infty$	2.4

Our conclusion from the analysis and experiments presented in this section is that *for zero order initial approximations the Richardson method in* RKR *mode is a rather efficient method for solving nonlinear systems of the type* (2.1), *and that the original Richardson method is less sucessful particularly for large values of* $\tau\sigma$.

4. SPECIFICATION OF THE TWO-LEVEL ALGORITHM

In Section 1.3 of the introduction we already described the two-level algorithm in the form

(4.1) *Prediction; (Coarse grid iteration; Fine grid iteration)*$^{\nu}$, $\nu \geq 0$.

Note that the fine grid iteration is always started with the initial approximation (1.18).

In our experiments we concentrated on the behaviour of the three basic parts in the process (4.1). We did not consider the effort of the generating implicit integration formula. All experiments were performed using backward Euler, i.e. (3.17) with $b_0 = 1$. The prolongator and restrictor were defined by (1.12) and (1.13), respectively.

The *coarse grid iteration* was defined by the process (we add an index h or H in order to indicate whether the iteration is performed on the fine grid or the coarse grid)

(4.2) $R_H(.1;.01)$, $H = 2h$,

the *fine grid iteration* by

$$(4.3) \qquad R_h(.22;.25).$$

The parameters D and α in (4.2) and (4.3) were determined on the basis of a large number of experiments which may be found in the appendix to [7].

The *predictor* was defined by

$$(4.4) \qquad RKR_H(.94);\ R_h(.22;.25),$$

where the initial approximation in the RKR_H process was simply

$$\hat{\vec{y}}_H = I_H^h \vec{y}_n$$

and in the R_h process the initial approximation was chosen according to (1.5'). The parameters D and α in (4.4) were again determined on the basis of numerical experiments (see appendix).

The two-level algorithm can now be written in the form

$$(4.5) \qquad RKR_H(.94);\ R_h(.22;.25)[R_H(.1;.01)R_h(.22;.25)]^{\nu},$$

where $\nu \geq 0$.

5. NUMERICAL EXPERIMENTS WITH TWO-LEVEL ALGORITHMS

In this section we test the two-level algorithm (4.5) and we try to get insight in the effect of the *prolongator*, the *predictor formula* and whether *Richardson iteration* is to be preferred when compared with *Gauss-Seidel iteration*. We also compare (4.5) with other one-grid methods.

In the tables of results we have used the following notations:

τ - integration step size

ν - number of coarse and fine grid iteration processes

$\Sigma\vec{f}_h$ - number of $\vec{f}_h$-evaluations on Ω_h

$\Sigma\vec{f}_H$ - number of $\vec{f}_H$-evaluations on $\Omega_H = \Omega_{2h}$

$\Sigma\vec{f} = \Sigma\vec{f}_h + \frac{1}{4}\Sigma\vec{f}_H$ - giving the total computational effort in terms of $\vec{f}$-evaluations on the fine grid.

A - accuracy defined by (3.20).

5.1. Effect of the prolongator

In Section 1.2 it was shown that the prolongator I_h^H and the restrictor I_H^h should be such that in actual computation the effect of $I_h^H I_H^h$ resembles the identity operator, otherwise the coarse grid correction is less efficient (cf. (1.16)). In order to see the behaviour of the prolongator defined by (1.12) we applied the algorithm (4.5) with $\nu = 0$ and listed the accuracy A obtained after the coarse grid process RKR_H, the coarse grid correction (1.5') and the smoothing process R_h. In Table 5.1 these results are given for problem (3.21) with several choices of the domain Ω. For the domain $0 \le x_1, x_2 \le 1$ the accuracy decreases considerably if the coarse grid

TABLE 5.1

A-values for problem (3.21) at t=1 with τ=1 and h = 1/20

	$0 \le x_1, x_2 \le 1$	$1 \le x_1, x_2 \le 2$	$5 \le x_1, x_2 \le 6$
$\vec{y}_H^{**}$	3.2	4.6	4.1
$\vec{y}_h = I_h^H \vec{y}_H^{**} + (I - I_h^H I_H^h)\vec{y}_h(0)$	1.0	4.0	4.1
$\vec{y}_h^{**}$	1.7	4.2	4.8

correction is introduced. This can be explained by considering the effect of the operator $I - I_h^H I_H^h$ on the function $\vec{y}_h(t)$ for t = 0 with components (cf. (3.21b))

$$(5.1) \qquad \left(\frac{4}{5}(ih + jh)\right)^{1/4}, \qquad i,j = 1,2,\ldots,h^{-1} - 1.$$

Since I_h^H is based on linear interpolation (cf. (1.12)), one can verify that $I_h^H I_H^h$ does not behave as the identity operator when applied to (5.1), particularly in the grid point $\vec{x} = (h,h)$. By shifting the domain Ω away from the origin, the operator $I_h^H I_H^h$ is applied to grid frunctions with bounded discrete first derivative as $h \to 0$ so that the linear interpolation behaves

adequately resulting in a small drop in accuracy.

5.2. The predictor formula

The two-level algorithm (4.5) uses a predictor which is based on the considerations in Section 3.2 where the approximation error is analyzed. At first sight, however, one might choose the following two-level algorithm

$$(5.2) \qquad [\mathcal{R}_H(.1;.01);\ \mathcal{R}_h(.22;.25)]^{\nu+1}, \qquad \nu \geq 0.$$

We want to show that the algorithm (4.5) is really a better choice than (5.2). We also listed the results obtained by the algorithm

$$(5.3) \qquad RKC;\ \mathcal{R}_h(.22;.25)[\mathcal{R}_H(.1;.01);\ \mathcal{R}_h(.22;.25)]^{\nu}, \qquad \nu \geq 0$$

where RKC denotes the application of the first order Runge-Kutta-Chebyshev method (cf. [6]) to the differential equation $\dot{\vec{y}}_H = \vec{f}_H(t,\vec{y}_H)$ on $\Omega_H = \Omega_{2h}$. The result obtained is then prolongated (of course without coarse grid correction) and smoothed by the $\mathcal{R}_h$ process. Thus, (5.3) differs from (4.5) only by using another predictor formula.

TABLE 5.2.
A-values for problem (3.21) at t=1 and
$\Omega = \{\vec{x} \mid 1 \leq x_1, x_2 \leq 2\}$ with h = 1/20.

τ	ν	$\Sigma\vec{f}^{(4.5)}$	A	$\Sigma\vec{f}^{(5.2)}$	A	$\Sigma\vec{f}^{(5.3)}$	A
	0	3+28	4.2	3+4	1.7	3+25	3.7
1	1	6+32	3.9	6+8	2.7	6+29	3.9
	2	9+36	4.0	9+12	3.8	9+33	4.0
	0	15+60	4.7	15+20	2.5	15+53	4.1
1/5	1	30+80	4.7	30+40	3.4	30+73	4.8
	2	45+100	4.8	45+60	4.6	45+93	4.8
	0	30+85	5.5	30+40	2.8	30+74	4.2
1/10	1	60+125	5.0	60+80	3.8	60+114	4.9
	2	90+165	5.1	90+120	5.0	90+154	5.1
	0	60+122	4.9	60+79	3.1	60+105	3.8
1/20	1	120+200	5.3	120+155	4.1	120+183	5.4
	2	180+279	5.4	180+236	5.4	180+262	5.4

In Table 5.2 the results are given obtained for problem (3.21). Obviously, the algorithm (4.5) is superior due to its more accurate predictor formula. On the other hand, the algorithm (5.2) is rather cheap if a modest accuracy is desired. However, if more rapid variations with t enter into the problem, the algorithm (5.2) tends to become unstable for larger integration steps. This may be concluded from Table 3.4 where the same problem was integrated, but on the domain $0 \le x_1, x_2 \le 1$ which results in a slightly stronger t-dependency. The algorithms (4.5) and (5.3) are less sensitive to t-variations because their predictor formulas are based on Runge-Kutta methods with many intermediate points so that actually rather small intermediate integration steps are performed.

5.3. Comparison with other methods

In order to get insight in the efficiency of the algorithm (4.5) we compared this method with two other integration techniques. The first one is of the form (5.2) but we replaced the Richardson iteration by Gauss-Seidel iteration which is defined as follows: let $\vec{y}^{(j)}$ be the j-th iterate with components $y_i^{(j)}$, i = 1,2,...r, then the components $y_i^{(j+1)}$ of the (j+1)-th iterate are determined by solving

$$(5.4) \qquad [L(y_1^{(j+1)},\ldots,y_{i-1}^{(j+1)}, y, y_{i+1}^{(j)},\ldots,y_r^{(j)})]_i = [\vec{\Sigma}]_i, \quad i = 1(1)r,$$

for y and setting $y_i^{(j+1)} = y$, i = 1(1)r. In (5.4) L and $\vec{\Sigma}$ correspond to the fine or the coarse grid problem (1.3a) or (1.3b) and $[\vec{v}]_i$ denotes the i-th component of a vector $\vec{v}$. One (nonlinear) Gauss-Seidel iteration is now defined by (approximately) solving (5.4) performing just one Newton iteration for each scalar equation. We applied the iteration process in a symmetric version by alternatingly reversing the order of the components in (5.4) [11]. This process will be denoted by $GS(m)$ where m denotes the number of iterations. The two-level algorithm using symmetric Gauss-Seidel iteration applied in our comparitive tests is given by

$$(5.5) \qquad [GS_H(14);\ GS_h(2)]^{\nu}, \qquad \nu > 0.$$

The number of Gauss-Seidel iterations in (5.5) were chosen such that it requires just the same number of $\vec{f}_h$ and $\vec{f}_H$ evaluations as required by the fine and coarse grid iterations in the algorithms (4.5), respectively.

TABLE 5.2

A-values for problem (3.21) at t=1 and
$\Omega = \{\vec{x} \mid 0 \leq x_1, x_2 \leq 1\}$ with h = 1/20

Richardson scheme (4.5)				Gauss-Seidel scheme (5.5)				*RKC* method [6]		
τ	ν	$\Sigma\vec{f}$	A	τ	ν	$\Sigma\vec{f}$	A	τ	$\Sigma\vec{f}$	A
1	0	26	1.7	1/5	1	34	1.7			
1	2	40	2.6	1/5	2	68	2.4			
1	3	47	3.0	1/5	3	101	2.9	1	163	3.0
				1/5	4	135	3.5	1/5	325	3.4
				1/5	5	169	3.8			
1/5	1	97	4.0	1/5	6	203	4.0	1/20	639	4.1

The second method is the *RKC* method already mentioned in Section 5.2.

For problem (3.21) with h = 1/20 the results obtained are listed in Table 5.2. These results were selected in such a way that the accuracies are more or less comparable while the $\Sigma\vec{f}$-value is minimized with respect to the set of (τ,ν) parameters. (This always occurred for the larger τ-values). The Richardson scheme (4.5) turns out to be cheaper than the Gauss-Seidel scheme (5.5) and the *RKC* scheme for comparable values of A. This is due to the possibility to integrate the whole integration interval in a single step, whereas the Gauss-Seidel scheme is unstable for such large steps. It should be observed, however, that this unstable behaviour for large steps can be avoided by solving the implicit equations (5.4) with more than one Newton step. These aspects are part of an iteration strategy for a general multigrid program and are not considered in this paper.

6. CONCLUDING REMARKS

We have investigated a two-level algorithm for solving the (non-linear) systems which arise when an implicit (one-step) integration method is applied to the semi-discretized form of a parabolic initial boundary value problem. The algorithm is based on a special form of the (nonlinear) Richardson iteration method. This algorithm has the following characteristics:

(i) The *storage requirements* are 5 arrays the dimension of which equals the number of grid points of the finest grid.

(ii) The algorithm allows rather *large integration steps* even in *strongly nonlinear problems*. This is achieved by the predictor formula based on the *RKR* mode of the Richardson method which constructs an initial approximation in a Runge-Kutta type fashion. If (nonlinear) Gauss-Seidel iteration is used the algorithm shows instabilities for large integration steps unless more Newton iterations are performed (and consequently additional right hand side functions are evaluated).

(iii) The computations only involve *right hand side evaluations* and the updating of the *spectral radius* of the Jacobian matrix (the right end point of the (negative) eigenvalue interval is not required). The Gauss-Seidel process does not require the spectral radius but the diagonal elements of the Jacobian matrix.

(iv) The algorithm is rather *flexible* in that respect that the damping of the higher and lower frequencies can be adjusted by monitoring the parameters α and D.

REFERENCES

[1] BRANDT, A., *Multi-Level Adaptive Techniques (MLAT) for Singular Perturbation Problems*, In: *Numerical Analysis of Singular Perturbation Problems*, P.W. Hemker and J.J.H. Miller (eds), Academic Press, London, 1979.

[2] FAIRWEATHER, G. & A.R. MITCHELL, *A new computational procedure for ADI-methods*, SIAM J. Numer. Anal. 4 (1967) pp. 163-170.

[3] FORSYTHE, G.E. & W.R. WASOW, *Finite difference methods for partial differential equations*, John Wiley & Sons, New York, (1960).

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

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

[6] HOUWEN, P.J. van der & B.P. SOMMEIJER, *On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values,* ZAMM 60 (1980) pp. 479-485.

[7] HOUWEN, P.J. van der & B.P. SOMMEIJER, *Analysis of Richardson iteration in multigrid methods for nonlinear parabolic differential equations,* Report NW 105/81, Mathematisch Centrum, Amsterdam (1981).

[8] HOUWEN, P.J. van der & H.B. de VRIES, *Preconditioning and coarse grid corrections in the solution of the initial value problems for nonlinear partial differential equations,* Report NW 95/80, Mathematisch Centrum, Amsterdam (1980), (to appear in SIAM J. on Sci. and Stat. Comp.).

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

[10] LINDBERG, B., *IMPEX - A program package for solution of systems of stiff differential equations,* Report NA 72.50, The Royal Institute of Technology, Stockholm (1972).

[11] ORTEGA, J. & W. RHEINBOLD, *Iterative solution of nonlinear equations in several variables,* Academic Press, New York (1970).

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

[13] SOMMEIJER, B.P., P.J. van der HOUWEN & J.G. VERWER, *On the treatment of time-dependent boundary conditions in splitting methods for parabolic differential equations,* Intern. J. Numer. Meth. Engin. 17 (1981) pp. 335-346.

[14] STETTER, H.J., *The defect correction principle and discretization methods,* Numer. Math. 29 (1978) pp. 425-443.

[15] VERWER, J.G., *An implementation of a class of stabilized explicit methods for the time-integration of parabolic equations,* ACM Trans. on Math. Software, (June 1980).

APPENDIX

In this appendix we explain our choice of the damping parameters D, D_h and D_H, and the α-parameters α_h and α_H in the two-level algorithm (cf. (4.5))

(A1) $$RKR_H(D)R_h(D_h;\alpha_h)[R_H(D_H;\alpha_H)R_h(D_h;\alpha_h)]^{\nu}, \qquad \nu \geq 0.$$

We describe a number of experiments for the nonlinear test problem (3.21) on the unit square $0 \leq x_1, x_2 \leq 1$ with $H = 2h = 1/10$. In the tables of results we use the same notations as in Section 5.

A1. The parameter D in the predictor formula

The computational effort involved in applying $RKR_H(D)$ increases if D is decreased (cf. (3.18")), therefore we were interested in the amount of damping needed to obtain an acceptable prediction for the fine grid solution. It turned out that hardly any damping is needed. In Table A1 some results are given obtained by the predictor formula

(A2) $$RKR_H(D)R_h(.22;.25)$$

for various values of D showing that even for extremely high damping the accuracy is not increased (the parameters in the fine grid iteration $R_h(.22;.25)$ do not essentially affect the behaviour of the predictor formula).

TABLE A1

A-values obtained at t=1 by scheme (A2)

		D = .94		D ≃ .1		D ≃ 10^{-3}		D ≃ 10^{-7}	
τ	$\Sigma\vec{f}_h$	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A
1	3	92	1.7	183	1.7	456	1.7	911	1.7
1/5	15	189	2.8	373	2.7	925	2.7	1845	2.7
1/10	30	268	3.9	526	3.9	1300	3.9	2590	3.9
1/20	60	385	4.0	750	4.9	1845	4.9	3670	4.9

Consequently we chose D = .94 in (A1).

A2. The fine grid parameters D_h and α_h

In order to avoid disturbing influences caused by an inaccurate solution of the coarse grid problem we applied (A1) with rather pessimistic values of D_H and α_H, i.e. we applied the algorithm

$$(A3) \qquad RKR_H(.94)R_h(D_h;\alpha_h)[R_H(.1;.01)R_h(D_h;\alpha_h)]^{\nu}, \qquad \nu \geq 0.$$

Since the fine grid iteration is relatively expensive we chose the pair $(D_h;\alpha_h)$ such that only two iterations were required. In Figure A1

TABLE A2

A-values obtained at t=1 by scheme (A3)

					$(D_h;\alpha_h)$				
τ	ν	$\Sigma\vec{f}_h$	$\Sigma\vec{f}_H$	$\Sigma\vec{f}$	(.06;.5)	(.22;.25)	(.29;.20)	(.38;.15)	(.5;.10)
	0	3	92	26	1.4	1.7	1.7	1.7	1.6
	1	6	108	33	1.8	2.1	2.2	2.3	2.2
	2	9	124	40	2.1	2.6	2.8	2.9	2.7
1	3	12	140	47	2.4	3.0	3.2	3.2	3.1
	4	15	156	54	2.7	3.2			3.2
	5	18	172	61	2.9				
	6	21	188	68	3.1				
	0	15	189	62	2.2	2.8	3.0	3.2	3.7
1/5	1	30	269	97	3.2	4.0	4.1	4.1	4.1
	2	45	349	132	4.1	4.1			
1/10	0	30	268	97	3.1	3.9	4.1	4.3	4.3
	1	60	427	167	4.4	4.4	4.4	4.4	4.4
1/20	0	60	385	156	3.9	4.0	4.0	4.0	4.0
	1	120	685	291	4.7	4.7	4.7	4.7	4.7

the eigenvalues corresponding to $R_h(D_h;\alpha_h)$ are illustrated in the case of a linear problem. These eigenvalues are given by (cf. (2.6))

$$P_2(z) = \frac{2(w_0+w_1-w_1 b_0 z)^2-1}{2w_0^2-1}, \tag{A4}$$

where z denotes the eigenvalues of the matrix $\tau J = \tau \partial \vec{f}/\partial \vec{y}$, w_0 and w_1 are defined in (2.3) and where the maximal value of $|P_2(z)|$ in $[-\tau\sigma_h, -\alpha_h\tau\sigma_h]$ is

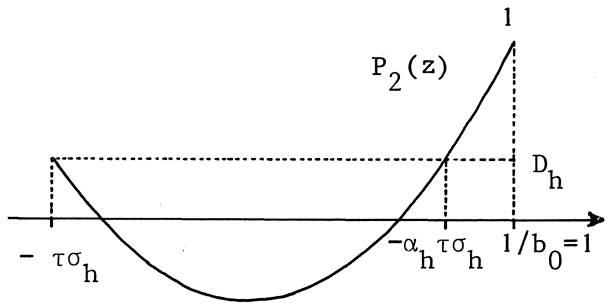

Fig. A1. Eigenvalues of $\mathcal{R}_h(D_h;\alpha_h)$ for linear problems

given by

$$D_h = [2(\frac{1+\alpha_h}{1-\alpha_h} + \frac{2}{b_0\tau\sigma_h(1-\alpha_h)})^2 - 1]^{-1}$$

$$\cong \frac{1-2\alpha_h+\alpha_h^2}{1+6\alpha_h+\alpha_h^2} \qquad \text{as } \tau\sigma_h >> 1 \text{ and } \alpha_h > 0. \tag{A5}$$

In Table A2 the results for several pairs $(D_h;\alpha_h)$ are given indicating that strong damping is less important than a sufficiently large interval where the damping is applied, except for small integration steps.

Furthermore, we note that it seems to be cheaper to use large integration steps combined with large ν-values than small integration steps and low values of ν.

It is also of interest to notice that for large τ-values the amount of work on the coarse grid is relatively large when compared with the work on the fine grid. As already mentioned in the introduction, this strongly suggests a recursive use of the coarse grid iteration technique.

In order to draw conclusions from the eigenvalue spectrum $P_2(z)$ illustrated in Fig. A1, we compare $P_2(z)$ with the eigenvalue spectrum of the iteration matrix in the Gauss-Seidel method (5.4) when applied to a linear problem with L given by the matrix $I - b_0\tau J$. By splitting J according to

$$J = J_1 + J_2 + J_3,$$

where J_1, J_2 and J_3 are respectively the lower triangle, the diagonal and the upper triangular matrix, we may write (5.4) in the vector form

$$(A6) \qquad -b_0\tau J_1 \vec{y}^{(j+1)} + [I - b_0\tau J_2]\vec{y}^{(j+1)} - b_0\tau J_3 \vec{y}^{(j)} = \vec{\Sigma} .$$

If we assume that the matrix $I - b_0\tau J$ has property (A) (see e.g. [3, p.243]) we may express the eigenvalues of the iteration matrix associated with (A6) in terms of those of the Jacobi method, i.e. in terms of the eigenvalues of the matrix

$$(A7) \qquad b_0\tau[I - b_0\tau J_2]^{-1}[J_1+J_3].$$

Let λ denote the eigenvalues of (A7) and $\tilde{\lambda}$ those of the Gauss-Seidel method (A6), then $\tilde{\lambda} = \lambda^2$ (cf. [3, p.250]). In order to express the eigenvalues λ in terms of the eigenvalues z of τJ we restrict our considerations to problems where J_2 is a scalar matrix $-dI$ with $d > 0$ (an example of such a problem was considered in Section 3.2, p.15). It is now easily verified that the eigenvalues of (A6) are given by

$$(A8) \qquad \eta(z) = \left[\frac{b_0(z+\tau d)}{1+b_0\tau d}\right]^2 .$$

The behaviour of $\eta^2(z)$ is illustrated in Figure A2 (two Gauss-Seidel iterations require roughly the same computational effort as two Richardson iterations if the implicit relations in (5.4) are solved in just one Newton iteration.

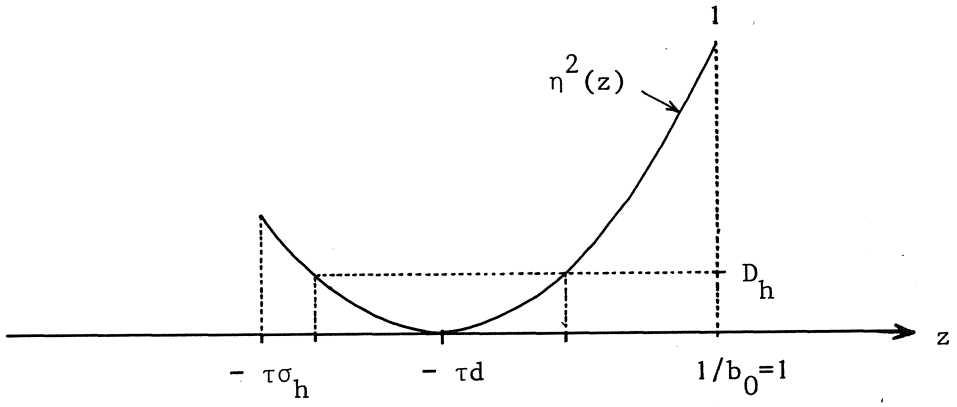

Fig. A2. Eigenvalues of (5.4) for linear problems

In the comparison of the spectrum functions $P_2(z)$ and $\eta^2(z)$ one should take into account that $P_2(z)$ corresponds to an eigenvector of τJ with eigenvalue z, whereas $\eta(z)$ usually corresponds to a completely different vector. For instance, if J is a normal matrix then the eigenvectors of the Richardson iteration matrix $P_2(\tau J)$ can be chosen orthogonally. The eigenvectors of the Gauss-Seidel iteration matrix are not necessarily orthogonal.

Since in the fine grid iteration our first interest is in the location of the intervals of high damping we consider the intervals on the z-axis where the eigenvalues are less than D_h in absolute value. For Richardson's method this interval is trivially given by $-\tau\sigma_h < z < -\alpha_h\tau\sigma_h$, and for Gauss-Seidel's method we find (after two iterations)

$$-\tau d - [\frac{1}{b_0} + \tau d]D_h^{1/4} < z < -\tau d + [\frac{1}{b_0} + \tau d]D_h^{1/4}.$$

In most practical cases these intervals just contain the eigenvalues corresponding to the eigenvectors of high frequency (e.g. in the case of the diffusion equation $u_t = \Delta u$). The length of these high frequency intervals is respectively given by $(1-\alpha_h)\tau\sigma_h$ and $\sim 2D_h^{1/4}\tau d$ where we have assumed that $1/b_0 << \tau d$. In the model problem $u_t = \Delta u$ discretized in the usual way on a uniform grid, we have $d = \sigma_h/2$ (observe that a necessary condition for the convergence of the Gauss-Seidel method (A6) requires that $d > \sigma_h/2 - 1/2b_0\tau$). Thus, Richardson's method has a larger interval of high damping (or

equivalently, has a higher damping in the same interval of high frequencies) if

$$(A9) \qquad (1-\alpha_h)^4 > D_h.$$

From (A5) it easily follows that this condition is satisfied for $\alpha_h \leq 1/2$.

The considerations above can be extended for the case where the number of fine grid iterations equals m. The intervals of high frequencies with damping $\leq D_h$ are then respectively given by $(1-\alpha_h)\tau\sigma_h$ and $\sim D_h^{1/(2m)}\,\tau\sigma_h$ where α_h and D_h are related by (cf. (2.7))

$$(A5') \qquad D_h \cong [\cosh m[\operatorname{arccosh} \frac{1+\alpha_h}{1-\alpha_h}]]^{-1}.$$

For example, if we choose $\alpha_h = 1/4$ then Richardson's method has a damping factor $\leq D_h$ in the interval of length $3\tau\sigma_h/4$ whereas the Gauss-Seidel method damps with a factor $\leq D_h$ in the interval of length $\sim \sqrt[2m]{2}\ \tau\sigma_h/3$.

Generally, we may conclude that Richardson's method has a stronger damping effect on the higher harmonics than the Gauss-Seidel method, in particular if the number of fine grid iterations is increased. In our experiments we chose m = 2, although a higher value of m might be more efficient in terms of computational effort versus accuracy. This aspect is part of a general iteration strategy and is not discussed in this paper. This choice of α_h is less critical as long as it is sufficiently small. We chose $\alpha_h = 1/4$ and consequently the damping factor D_h becomes $\sim .22$.

A3. The coarse grid parameters D_H and α_H

The only parameters left in the algorithm (A1) are D_H and α_H. We applied the algorithm

$$(A10) \qquad RKR_H(.94)R_h(.22;.25)[R_H(D_H;\alpha_H)R_h(.22;.25)]^{\nu}, \qquad \nu \geq 0$$

for a large number of $(D_H;\alpha_H)$-value revealing that the accuracy is largely independent of D_H and α_H. In Table A3 a few results are listed for $D_H = .1$. These results suggest to choose $\alpha_H \geq .05$ in order to increase the efficiency.

TABLE A3

A-values obtained at t=1 by scheme A(10) with $D_H = .1$

			$\alpha_H = .05$		$\alpha_H = .01$		$\alpha_H = .001$	
τ	ν	$\Sigma\vec{f}_H$	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A
	0	3	92	1.7				
	1	6	100	2.1	108	2.1	139	2.1
1	2	9	108	2.6	124	2.6	186	2.6
	3	12	116	3.0	140	3.0	233	3.0
	4	15	124	3.2	156	3.2	280	3.2
	0	15	189	2.8				
1/5	1	30	229	4.1	269	4.0	389	4.1
	2	45	269	4.4	349	4.1	589	4.1
	3	60	309	4.3				
	0	30	268	3.9				
1/10	1	60	348	5.3	427	4.4	616	4.4
	2	90	428	4.8	586	4.4	964	4.4
	3	120	508	4.6				
	0	60	385	4.0				
1/20	1	120	545	4.9	685	4.7	953	4.8
	2	180	707	5.1	985	4.8	1521	4.8

However, by using larger α_H-values less low frequencies are damped which would violate the general idea that the low frequencies should be removed on the coarse grid and the high frequencies on the fine grid (cf. the discussion in §1.4). Therefore, we choose the safe value $\alpha_H = .01$.

By the same arguments one should choose D_H sufficiently small and one may ask whether $D_H = .1$ yields a sufficient reduction of the lower frequencies. It turns out, however, that the accuracy is more or less insensitive to changes in the damping parameter D_H, in particular for larger integration steps. Even for rather low damping the accuracy does not decrease as can be seen in Table A4. It is tempting to choose $D_H > .1$ which again saves $\vec{f}_H$-evaluations, but we decided to use a safe damping parameter and put $D_H = .1$ in all experiments reported in §5.

TABLE A4

A-values obtained at t=1 by scheme (A10) with $\alpha_H = .01$

			$D_H = .33$		$D_H = .8$		$D_H = .99$	
τ	ν	$\Sigma\vec{f}_h$	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A	$\Sigma\vec{f}_H$	A
	0	3	92	1.7				
	1	6	102	2.1	97	2.2	95	2.2
	2	9	112	2.6	102	2.6	98	2.6
	3	12	122	3.0	107	3.0	101	3.0
1	4	15	132	3.2	112	3.3	104	3.2
	6	21			122	3.2	110	3.6
	7	24					113	3.8
	9	30					119	3.7
	0	15	189	2.8				
	1	30	239	4.1	214	4.1	204	4.1
1/5	2	45	289	4.1	239	4.5	219	4.6
	3	60			264	4.3	234	4.9
	9	150					324	4.4
	0	30	268	3.9				
	1	60	368	4.4	318	4.9	298	4.5
1/10	2	90	468	4.4	368	4.8	328	4.7
	4	150					388	5.3
	9	300					538	4.7
	0	60	385	4.0				
	1	120	575	4.7	485	4.8	445	4.2
	2	180	765	4.8	585	5.2	505	4.4
	3	240			685	4.9	565	4.6
1/20	4	300			785	4.8	625	4.9
	6	420					745	5.5
	9	600					925	5.0