stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NN 22/80 DECEMBER

E.J. VAN ASSELT

A SURVEY OF MULTI-GRID METHODS FOR NONLINEAR PROBLEMS

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: 65N20, 65R20, 65F10, 65J15

A survey of multi-grid methods for nonlinear problems

by

E.J. van Asselt

ABSTRACT

In this report different approaches to solve nonlinear problems by multi-grid methods are described. In its elementary form a multi-grid method is a recursive application of the two level algorithm which itself consists of the iterative use of two procedures:

a. a smoothing procedure on the fine grid

b. a coarse grid correction.

These procedures can all be cast into the form of a defect correction process.

Attention is payed to the determination of an initial approximation for the iterative process: by an imbedding method, by the full multi-grid method or by a combination of both.

Two well known variants of the full multi-grid method are described. Finally a method for the determination of the coarsest and the finest grid size is given.

KEY WORDS & PHRASES: *Defect correction process, multi-level algorithm, imbedding method, full multi-grid method*

CONTENTS

1. A DEFECT CORRECTION PROCESS FOR NONLINEAR EQUATIONS

Let A,B be normed linear spaces, and M be a linear or non-linear mapping from A to B. Consider the equation

(1.1) $\quad Mu = g.$

Assume that $g \in Y$ and that for $X \subset A$ and $Y \subset B$ the mapping $M: X \to Y$ is surjective.
Assume that for some subset $\tilde{Y} \subset Y$ with $\tilde{g} \in \tilde{Y}$ a continuous mapping $\tilde{G}: \tilde{Y} \to X$ exists such that $\tilde{G}\tilde{g}$ is an approximation to a solution of (1.1).
$\tilde{G}$ is called an approximate inverse of M. If $\tilde{G}$ is injective, then $\tilde{M}: \tilde{G}\tilde{Y} \to \tilde{Y}$ exists, such that $\tilde{M}\tilde{G} = I_{\tilde{Y}}$, with $I_{\tilde{Y}}$ the identity operator on $\tilde{Y}$; i.e. $\tilde{M}$ is the left inverse of $\tilde{G}$. $\tilde{M}$ is called an approximation to M. Thus an approximation of (1.1) is given by

(1.2) $\quad \tilde{M}\tilde{u} = \tilde{g}; \quad \tilde{g} \in \tilde{Y} \subset Y.$

Many iterative methods to solve (1.1) can be cast into the form of the following defect correction process (DCP):

(1.3)
$$u_0 = \tilde{G}_0 g$$
$$u_{i+1} = u_i + \mu\tilde{G}_{i+1}(\tilde{g}+(g-Mu_i)/\mu)-\mu\tilde{G}_{i+1}\tilde{g}, \quad i = 0,\ldots$$

$\tilde{G}_i$ is an approximate inverse of M, and μ and $\tilde{g}$ are free to choose (HEMKER [6]). If $\tilde{G}_i = \tilde{G}$ for all i, we have a stationary DCP, otherwise the process is called non-stationary.

REMARK. In case of linear operators $\tilde{G}_i$, (1.3) reduces to

(1.4)
$$u_0 = \tilde{G}_0 g$$
$$u_{i+1} = u_i + \tilde{G}_{i+1}(g-Mu_i)$$

M still may be nonlinear.

In case of injective $\tilde{G}_i$, we can introduce the approximate operators $\tilde{M}_i$ and write (1.4) as a sequence of equations:

$$\tilde{M}_0 u_0 = g$$
$$\tilde{M}_{i+1} u_{i+1} = \tilde{M}_{i+1} u_i + g - Mu_i. \tag{1.5}$$

EXAMPLE

(1.6) Newton iteration

$$u_{i+1} = u_i + (M'(u_i))^{-1}(g-Mu_i).$$

Here we have a non-stationary DCP with $\tilde{G}_{i+1} = \{M'(u_i)\}^{-1}$, where $M'(u_i)$ is the Fréchet-derivative of M at the point u_i. All $\tilde{G}_i$ are linear operators.

2. MULTIGRID METHODS FOR NONLINEAR PROBLEMS

2.0. Introduction

Discretization of (1.1) on a given uniform grid G_h with mesh size h yields the discrete form

$$M_h u_h = g_h \quad \text{where } M_h: X_h \to Y_h \text{ is surjective by assumption.} \tag{2.0.1}$$

The (non)linear multigrid method (MGM) solves (2.0.1), using discretizations on coarser grids G_H, denoted by

$$M_H u_H = g_H. \tag{2.0.2}$$

In its elementary form, the MGM is a recursive application of the two level algorithm (TLA) which itself consists of the iterative use of two procedures:

a. A smoothing procedure on the fine grid
b. A coarse grid correction.

In section 2.1 we treat the smoothing process and we give 4 examples of it.

In section 2.2 we introduce restrictions and prolongations and consider

coarse grid corrections.

In section 2.3 we describe the two level algorithm TLA in an Algol-like program.

In section 2.4 we give the recursive application of the TLA in a multilevel algorithm MLA.

2.1. The Smoothing Process

The smoothing process is a DCP or a combination of defect correction processes (see (2.1.1)). It is a step in the iterative solution of (2.0.1) which damps out the rapid fluctuations in the residulas so that they can be represented on a coarser grid.

The effect of the usual relaxation methods for the solution of difference equations as smoothing process can be analysed with discrete fourier analysis cf. BRANDT [1], DAHLQUIST & BJÖRCK [2], HEMKER [5].

EXAMPLES

(2.1.1) The nonlinear Gauss-Seidel process

Consider a system of n nonlinear equations. $\bar{f}(\bar{x}) = \bar{0}$ with $\bar{f} = (f_1,\ldots,f_n)$; $\bar{x} = (x_1,\ldots,x_n)$. Let $\bar{x}^{(0)}$ be an initial approximation to the solution. For the nonlinear Gauss-Seidel process $x_i^{(k+1)}$ is obtained by solving for x_i the i-th equation:

$$f_i(x_1^{(k+1)},\ldots,x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)},\ldots,x_n^{(k)}) = 0.$$

Generally, this is a nonlinear equation in x_i, and we can find an approximation $\tilde{x}_i$ to x_i, e.g. with Newton iteration. Finally we set $x_i^{(k+1)} = \tilde{x}_i$ (RHEINBOLDT [8]).

Thus the nonlinear Gauss-Seidel process with Newton iteration for a system of n nonlinear equations is an iterative process in which each iteration step consists of n non-stationary defect correction processes (Newton iteration)

$$\bar{x}_j^{(i+k/n)} = \bar{x}_{j-1}^{(i+k/n)} - (\delta_{1,k},\ldots,\delta_{k-1,k},\tilde{G}_{j-1}^{(i+k/n)} f_k(\bar{x}_{j-1}^{(i+k/n)}),$$

$$\delta_{k+1,k},\ldots,\delta_{n,k}); \quad j = 1,\ldots,N_{i+k/n}$$

$$\bar{x}_0^{(i+k/n)} = \bar{x}_{N_{i+(k-1)/n}}^{(i+(k-1)/n)}$$

$$k = 1,\ldots,n \qquad i = 0,\ldots .$$

$\delta_{.,.}$ is the Kronecker delta, $\delta_{i,j} = 0$ for $i \neq j$, $\delta_{i,j} = 1$ for $i = j$. $\tilde{G}_{j-1}^{(i+k/n)} = (d_k f_k\ \bar{x}_{j-1}^{(i+k/n)})^{-1}$, the inverse of the Fréchet-derivative to the k-th variable of f_k at $\bar{x}_{j-1}^{(i+k/n)}$.

In general the number of steps: $N_{i+k/n}$ of each non-stationary defect correction process can be determined during the actual computation by terminating the iterations depending on the speed of convergence. We can also devise a nonlinear Gauss-Seidel process where (2.1.1.1) is applied with only a fixed number of steps ($N_{i+k/n}$ is a constant for each i, k, e.g. $N_{i+k/n} = 1$).

(2.1.2) The linear Gauss-Seidel process

For a system of n linear equations Au = f (A a square n × n matrix) the linear Gauss-Seidel process is given by

$$u_{i+1} = u_i + L^{-1}(f - Au_i). \qquad \text{(See (1.4))}.$$

Here we have a stationary DCP with linear $\tilde{G} = L^{-1}$, where A = L + U, L lower triangular matrix and U strict upper triangular.

(2.1.3) Newton iteration and linear Gauss-Seidel

For a system of n nonlinear equations we can apply Newton iteration. In each iteration step we have a linear equation on which we can apply linear Gauss-Seidel iteration.

Newton:

$$\bar{u}_{i+1} = \bar{u}_i - (d\bar{f}(\bar{u}_i))^{-1}\ \bar{f}(\bar{u}_i) \quad \text{where } d\bar{f}(\bar{u}_i)$$

is the Jacobian matrix. This can be considered as a linear system (cf. (1.5)) with unknown $\bar{u}_{i+1}$:

$$(d\bar{f}(\bar{u}_i))\bar{u}_{i+1} = (d\bar{f}(\bar{u}_i))\bar{u}_i - \bar{f}(\bar{u}_i),$$

on which linear Gauss-Seidel can be applied.

(2.1.4) Smoothing by application of an integral operator

Consider the following (possibly nonlinear) equation in operator notation:

(2.1.4.1) $u - \tilde{K}u = f.$

If $\tilde{K}: A \to B = A$ is sufficiently smooth, an efficient smoothing process is the DCP:

(2.1.4.2) $u := \tilde{K}u + f.$

This means $\tilde{g} = g = f$; $M = I - \tilde{K}$, $\mu = -1$ and $\tilde{G}_{i+1} = I$ in (1.3).

As example of the smoothing effect consider:

$$(2.1.4.3)\quad u(y) - \int_{-\pi}^{\pi} K(x,y)u(x)dx = 1$$

with

$$K(x,y) = \sum_{k=1}^{\infty} \frac{\cos k(x-y)}{4k}, \quad -\pi \le x,y \le \pi; \quad u[-\pi] = u[\pi] = 1.$$

Let u_0 be an approximation to the exact solution $u \equiv 1$ on $[-\pi,\pi]$.
Assume that

a. u_0 is continuously differentiable on $[-\pi,\pi]$ except possibly in a finite number of points p.
b. in each such point p, the lower and upper limit of u_0 and u_0' exists, and $u_0(p) = \frac{1}{2}[\lim_{x\uparrow p} u_0(x) + \lim_{x\downarrow p} u_0(x)]$,

then on $[-\pi,\pi]$, u_0 can be written as:

$$(2.1.4.4)\quad u_0(x) = a_0 + \sum_{n=1}^{\infty} (a_n \sin nx + b_n \cos nx).$$

Suppose u_0 is chosen such that $a_0 = 1$.

One smoothing step yields:

$$u_1 = \int_{-\pi}^{\pi} K(x,y)u_0(x)dx + 1 = 1 + \sum_{k=1}^{\infty} (\frac{\pi}{4k} a_k \sin ky + \frac{\pi}{4k} b_k \cos ky).$$

The residual R_0 corresponding to u_0 is given by

$$R_0 = u_0 - \tilde{K}u_0 - 1 = \sum_{n=1}^{\infty} (\tilde{a}_n \sin nx + \tilde{b}_n \cos nx), \quad \text{with}$$

$$\tilde{a}_n = (1 - \frac{\pi}{4n}) a_n \quad \text{and} \quad \tilde{b}_n = (1 - \frac{\pi}{4n}) b_n.$$

The residual R_1 corresponding to u_1 is given by

$$R_1 = u_1 - \tilde{K}u_1 - 1 = \sum_{n=1}^{\infty} (\frac{\pi}{4n} \tilde{a}_n \sin nx + \frac{\pi}{4n} \tilde{b}_n \cos nx).$$

So $\tilde{a}_n$ and $\tilde{b}_n$ are reduced by a factor $\frac{\pi}{4n}$.
Thus all frequency components are damped. The high frequency components are damped more than the low frequency components.

Equation (2.1.4.1) can also be obtained by reformulation of a differential equation. E.g. consider the nonlinear problem:

P_1: $Lu = f(u,x)$, with L a linear second order elliptic differential operator. Replacing u by some known v we obtain a linear problem:

$$Lw = f(v,x).$$

The mapping $v \to w$ defines a nonlinear integral operator $K: A \to A$; $K(v) = L^{-1}f(v,x)$.
With this operator K, the integral-form of P_1 reads:

$$P_2: (I-K)u = 0.$$

The discretization of P_2 yields the discrete operator K_ν and we have to solve

$$(I-K_\nu)u_\nu = f_\nu.$$

The smoothing process is here

$$u_\nu^{(i+1)} = K_\nu(u_\nu^{(i)}) + f_\nu.$$

2.2. Coarse grid corrections

(2.2.1) Restrictions and prolongations

Let $\Omega \subset \mathbb{R}^k$, $k \in \mathbb{N}$.

DEFINITIONS.

a. A *grid* G is a set $\{x \mid x \in \Omega, x \text{ isolated}\}$.

b. Let $h = (h_1,\ldots,h_k)$, $h_i \in \mathbb{R}_{>0}$, $i = 1,\ldots,k$.
A *uniform grid* G_h with *mesh size* h is a set

$$\{(x_1,\ldots,x_k) \in \Omega \mid x_i/h_i \in \mathbb{Z}, \quad i = 1,\ldots,k\}$$

c. A *grid-function* u is a mapping $u: G \to \mathbb{R}$.

d. Let $G_H \subset G_h$ and let V_H, V_h be the linear spaces of grid-functions defined on G_H, G_h.
A *restriction* $R_{H,h}$ is a surjective linear mapping $R_{H,h}: V_h \to V_H$.
A *prolongation* $P_{h,H}$ is an injective linear mapping $P_{h,H}: V_H \to V_h$.
If we identify the spaces V_h with the normed spaces X_h or Y_h (see eq. (2.0.1)) then the corresponding restrictions and prolongations are denoted by R_{Hh} and P_{hH} (or $\bar{R}_{Hh}$ and $\bar{P}_{hH}$ respectively).
If no confusion is possible we simply denote the restrictions and prolongations by R, $\bar{R}$ and P, $\bar{P}$.

EXAMPLES

(2.2.1.1) Natural restriction

Let $G_H \subset G_h$ then the natural restriction $R: V_h \to V_H$ is defined by $Ru_h = v_H$ where $v_H(x) = u_h(x)$ $\forall x \in G_H$.

(2.2.1.2) Weighted restriction

Assume $G_H \subset G_h$ with uniform grids G_H, G_h, $\Omega = \mathbb{R}^k$.

The stencil B_h is a finite subset of G_h with $0 \in B_h$ together with a set of real values $\{\alpha_y \in \mathbb{R} \mid y \in B_h\}$.
The weighted restriction $R: V_h \to V_H$ is defined by $Ru_h = v_H$ with
$v_H(x) = \left(\sum_{y \in B_h} \alpha_y u_h(x+y)\right) / \left(\sum_{y \in B_h} \alpha_y\right)$.

(2.2.1.3) Prolongation by linear interpolation
Consider $\Omega = [0,1] \subset \mathbb{R}$, $h = \frac{1}{2n}$, $n \in \mathbb{N}$

$$G_h = \{x \mid x = \frac{k}{2n}, \quad k = 0,\ldots,2n\}$$

$$G_{2h} = \{x \mid x = \frac{k}{n}, \quad k = 0,\ldots,n\}.$$

The prolongation by linear interpolation is the mapping $P: V_{2h} \to V_h$ defined by $Pu_h = v_{2h}$, with

$$v_{2h}(\frac{k}{n}) = u_h(\frac{k}{n}) \qquad k = 0,\ldots,n$$

$$v_{2h}(\frac{2k+1}{2n}) = (u_h(\frac{k}{n}) + u_h(\frac{k+1}{n}))/2 \qquad k = 0,\ldots,n-1.$$

(2.2.2) The coarse grid correction as DCP
The coarse grid correction in a two level algorithm is a DCP given by:

$$u_h := u_h + \mu P(u_H - \tilde{u}_H)$$

with

$$u_H = M_H^{-1}(M_H(\tilde{u}_H) + (\bar{R}g_h - \bar{R}M_h u_h)/\mu)$$

where $\tilde{u}_H$ depends on the method used (see examples in paragraph 2.4).
Here $\tilde{G}_{i+1} = PM_H^{-1}\bar{R}$, and $\tilde{g}$ is such that $\bar{R}\tilde{g} = M_H\tilde{u}_H$.

2.3. The nonlinear two level algorithm (TLA)

One step of the two level algorithm TLA consists of p smoothing steps, followed by a coarse-grid correction step and another q smoothing steps. This TLA-step can be described in the following Algol-like program:

```
proc   tla = (ref gridf u, gridf g) void:

# one TLA iteration step in the solution of M_h u = g #

begin
      to p
      do smooth (u.g) od;
      d := M_H ũ_H + R̄(g-M_h u)/μ;
      v := ũ_H;
      # ũ_H is some approximation to the solution on level H #
      solve (v,d);
      # i.e. find the solution of M_H v = d with initial approxi-
        mation ũ_H #
      u := u + μP(v-ũ_H);
      to q
      do smooth (u,g) od
end;
```

2.4. The nonlinear multi-level algorithm (MLA)

The multi-level algorithm (MLA) is much similar to the TLA: only the exact solution on the coarser grid is replaced by a finite number of MLA iteration steps on this coarser grid.
In this way a recursive procedure is obtained. One step of the MLA is given in the following Algol-like program:

(2.4.1)

```
proc mla=(ref gridf u, gridf g) void:
      # one MLA iteration step in the solution of M_h u = g #
begin
      to p
      do smooth (u,g) od;
      d := M_H ũ_H + R̄(g-M_h u)/μ;
      v := ũ_H;
      if level of u = 0
      then solve (u,g) # on the coarsest grid #
      else to σ
           do mla (v,d) od
```

```
            fi;
            u := u + μP(v-ũ_H);
            to q
            do smooth (u,g) od
        end;
```

This algorithm has a fixed strategy, i.e. the numbers p,σ and q are fixed and independent of the course of the computation. If we introduce conditions in the algorithm depending on the actual course of the computation, e.g. iterations are terminated depending on the speed of convergence, the algorithm has an adaptive strategy.

EXAMPLES

(2.4.2) The FAS-algorithm

We obtain the FAS(= Full approximation storage) algorithm of Brandt [1] by selecting $\tilde{u}_H = Ru_h$ and $\mu = 1$ in (2.4.1).

(2.4.3) A nonlinear multilevel algorithm for integral equations

In [4] Hackbusch describes a multilevel algorithm which solves the nonlinear integral equation $(I-K)u = 0$. (See example (2.1.4)).
Discretization on a given grid G_h with meshsize h yields the discrete form:

$$(I-K_h)u_h = 0.$$

Hackbusch's algorithm is a special case of (2.4.1) in which $M_H = I - K_H$ and $M_h = I - K_h$, $\sigma = p = 1$; $q = 0$ and $\mu = -1$.
$\tilde{u}_H$ is an approximate solution of $(I-K_H)u_H = 0$ and the smoothing step reads:
$u_h := K_h u_h + g_h$ ($g_h = 0$ on the finest grid).

3. THE LINEAR MULTIGRID ALGORITHM

If the (nonlinear) multi-level algorithm is applied to a linear equation it reduces to a simpler form. For any choice of $\tilde{u}_H$ the algorithms are mathematically equivalent (see (3.3)).
Two formulations of the linear multigrid algorithm are known in the literature.

One, where in the coarse grid correction the *correction* is found as the solution of the linear system (correction storage (CS) - algorithm), and the other where the *corrected approximation* is found as the solution of the linear system (Full approximation storage (FAS)-algorithm).

3.1. The CS-algorithm (BRANDT [1])

M is linear and so are M_H and M_h.
Here one takes $\tilde{u}_H = 0$, $\mu = 1$.

3.2. The FAS-algorithm (BRANDT [1]).

This algorithm is the linear version of the algorithm (2.4.2), i.e. $\tilde{u}_H = Ru_h$, $\mu = 1$.

3.3. Comparision of the CS- and FAS algorithm

Mathematically the CS-algorithm (3.1) and the FAS-algorithm (3.2) are equivalent. For CS the coarse grid equation reads:

$$(3.3.1) \qquad M_H v_H = \bar{R}(g_h - M_h u_h)$$

with initial approximation $v_H = 0$, and for the approximation to the fine grid solution we have

$$(3.3.2) \qquad u_h := u_h + P(v_H).$$

For the FAS-algorithm the coarse grid equation reads:

$$(3.3.3) \qquad M_H u_H = M_H R u_h + \bar{R}(g_h - M_h u_h)$$

with initial approximation $u_H = Ru_h$. The approximation to the fine grid solution is given by

$$(3.3.4) \qquad u_h := u_h + P(u_H - Ru_h).$$

The equivalence of the CS- and FAS-algorithm is clear: Since M_H is linear,

(3.3.3) can be written as:

$$M_H(u_H - Ru_h) = \bar{R}(g_h - M_h u_h)$$

with initial approximation

$$u_H = Ru_h.$$

With $v_H = u_H - Ru_h$ we obtain equation (3.3.1) with initial approximation $v_H = 0$ and the fine grid approximation (3.3.2).

The only difference of the CS- and the FAS-formulation for linear problems lies in the implementation of the algorithms. We see that the CS-formulation is not applicable to nonlinear problems.

4. METHODS FOR THE DETERMINATION OF INITIAL APPROXIMATIONS

4.0. Introduction

Let $M(u) = 0$ be a nonlinear equation in a Banach space A. If we want to solve this equation with an iterative method, we need an initial approximation u^0 in a sufficiently small neighbourhood of the solution $\bar{u}$. To overcome the disadvantage of a direct guess of such an initial approximation we could apply an imbedding method (paragraph (4.1)), a full multigrid method (paragraph (4.2)) or a combination of both (paragraph (4.3)).

4.1. The imbedding method

Consider a family of problems $H(u,t) = 0$ $t \in [0,1]$ which is defined by a one-parameter *imbedding function* or *homotopy* $H: A \times [0,1] \to A$ such that

$$H(u_0, 0) = 0$$

for a certain known $u_0 \in A$

$$H(u,1) = M(u) \quad \text{for all } u \in A.$$

Assuming that the solution $u(t)$ of $H(u(t),t) = 0$ is continuously dependent on t, $u(t)$ is probably a sufficiently close approximation to $u(t+\Delta t)$, with Δt small. Thus the solution $u(1) = \bar{u}$ is obtained from $u(0) = u_0$ by continuous transition.

In numerical applications a finite sequence of problems $M(u,t_n) = 0$ $0 = t_0 < t_1 < \ldots < t_n = 1$ is considered. The transition $u(t_n) \to u(t_{n+1})$ can be done by the iterative solution of $M(u,t_{n+1}) = 0$ with initial approximation $u(t_n)$. An *imbedding method* is also called a *homotopy method*, or a *continuation method*. (cf. WACKER [9]).

EXAMPLES

(4.1.1) Defect reducing homotopy (WACKER [9])

$$H(u,t) = M(u) - (1-t)M(u_0).$$

(4.1.2) Regularizising homotopy (WACKER [9])

$$H(u,t) = tM(u) + (1-t)(u-u_0).$$

(4.1.3) Parameter continuation (POLAK, WACHTERS, BEELEN, HEMKER [7])
Consider the nonlinear (discrete) equation $M(u;\varepsilon) = 0$ with some parameter $\varepsilon \in \mathbb{R}$. Choose $M(u,t;\varepsilon) = M(u,\varepsilon(t))$ such that $M(u,\varepsilon(1)) = M(u,\varepsilon)$ and $M(u,\varepsilon(0))$ is easy to solve.

4.2. The full multigrid method (FMGM)

In order to solve the discretization of a nonlinear problem $Mu = 0$ on a (fine) grid G_N, in the FMGM a coarse to fine sequence of grids is used: $G_0, G_1, \ldots, G_N$. On each grid G_k with mesh size h_k we have the discrete nonlinear problem

$$P_k: M_{h_k} u_{h_k} = 0$$

(e.g. a differential or an integral form).

P_0 is solved first on the coarsest grid G_0, where only a small system

of equations has to be solved. Subsequently the problem is discretized on finer grids, where the prolongation of the coarse grid solution (of P_k) yields the initial approximation to the finer grid solution (of P_{k+1}). Each P_k is solved by an iterative method. (any method, e.g. Newton, a MLA, a combination of both).

NOTE. If the FMGM is used in combination with a MLA, (see 2.4.1), an approximation $\tilde{u}_H$ to the solution on level H, is already available from the previous step in the FMGM. (see also 2.4.3).

4.3. A combination of imbedding and FMGM

Consider the equation $M_{h_N} u_{h_N} = 0$ on the finest grid G_N. We can combine imbedding and FMGM in the following way by considering the sequence of problems:

$$P_{k,j}: \quad M_{h_k}(u_{h_k}, t_k^j) = 0 \qquad j = 0,1,\ldots,j_k$$

$$t_0^0 = 0 \le t_0^1 \le \ldots \le t_0^{j_0} \le \ldots \le t_k^0 \le t_k^1 \le \ldots \le$$

$$\le t_k^{j_k} \le \ldots \le t_N^{j_N} = 1.$$

(This means that on one grid more parameter-values of t are possible, and for one parameter-value of t, more grids can be used) such that

$$M_{h_0}(u_{h_0}, t_0^0) = 0$$

is easy to solve, and

$$M_{h_N}(u_{h_N}, t_N^{j_N}) = M_{h_N} u_{h_N}.$$

Each $P_{k,j}$ is solved with some iterative method with as the initial approximation either the solution of $P_{k,j-1}$ (if $j \ge 1$) or (if $j=0$) the prolongation of the solution of $P_{k-1,j_{k-1}}$.

5. LINEARIZATION OF NONLINEAR PROBLEMS

Two well known variants of the FMGM for nonlinear problems are:
(5.1) Linearization of the nonlinear equation on each grid G_k, (e.g. with Newton) and application of a linear MLA to the linearized equation (HACKBUSCH [3]) (i.e. in Newton iteration on the grid G_k, each iteration step means the solution of a linear equation which is done by one or more steps of the MLA).

(5.2) Application of a nonlinear MLA on each grid G_k.

Of course in the application of MLA linearization is not necessary. However there are many applications in which linearization occurs in the smoothing process:

EXAMPLES

(5.2.1) In the nonlinear Gauss-Seidel process Newton iteration can be applied (2.1.1).
(5.2.2) In examples (2.4.2) smoothing can be done by first Newton linearization and then Gauss-Seidel iteration.
(5.2.3) The assignment $u_h := K_h u_h + g_h$ (example 2.4.3) means the solution of a linear equation.

6. THE COARSEST AND THE FINEST GRID IN A FMGM

In this section we point out how the coarsest grid-size and the finest grid-size can be found. For a combination of both in one automatic program see HACKBUSCH [4].

6.1. The coarsest grid-size

The coarsest grid-size h_0 in a FMGM must be small enough, so that the corresponding discretization yields a sufficiently accurate approximation to the solution, and the iterative method to solve the problem on the grid with this grid-size converges to the required solution.
On the other hand, if we want to find an initial approximation for the full multigrid method as cheap as possible, the coarsest grid-size h_0 must be as

coarse as possible.

The optimal course grid-size can be found in the following way:

a) Choose the coarse grid G_0 with mesh size h_0.
b) Investigate the convergence of the iterative process on the next grid G_1.
c) In case of divergence start the full multigrid process again on a coarse grid $\bar{G}_0$ with mesh size $\bar{h}_0 < h_0$, and apply the same procedure.

6.2. The finest grid-size

6.2.1. DEFINITION. Let $M_h u_h = 0$ a discretization of the nonlinear equation $Mu = 0$.
Let $\|\cdot\|$ be a suitable norm, then the *discretization error* is defined by $\|u-u_h\|$.

The discretization of a continuous equation on a finer grid makes sense only as long as an approximation of the same accuracy cannot be obtained by a simple interpolation of a coarser grid solution.
So if we estimate the discretization error by $\|u_{h_k} - \tilde{P}u_{h_{k-1}}\|$, with $\tilde{P}$ a sufficiently accurate interpolation operator, and this error is small enough, we need not to continue the full multigrid method on smaller grids.

ACKNOWLEDGEMENT

The author wishes to thank Dr. P.W. Hemker (MC-Amsterdam) for his valuable advice.

REFERENCES

[1] BRANDT, A., *Multi-level adaptive solutions to boundary-value problems,* Mathematics of computation, volume 31, number 138, april 1979, pages 333-390.

[2] DAHLQUIST, G., Å. BJÖRCK, *Numerical methods,* Prentice-Hall, series in automatic computation.

[3] HACKBUSCH, W., *On the multi-grid method applied to difference equations,* Computing 20, 291-306 (1978).

[4] HACKBUSCH, W., *On the fast solutions of nonlinear elliptic equations,* Numerische Mathematik, 32, 83-95 (1979).

[5] HEMKER, P.W., *The incomplete LU-decomposition as a relaxation method in multigrid algorithms, Boundary and Interior Layers-Computational and Asymptotic Methods* (J.J.H. Miller ed.). Boole Press, Dublin, 1980.

[6] HEMKER, P.W., *Introduction to multigrid methods,* (Submitted for publication in: Nw. Arch. Wisk.).

[7] POLAK, S.J., A. WACHTERS, Th. BEELEN and P.W. HEMKER, *A mesh-parameter-continuation method,* NW-Report NW 89/80 MC; Department of numerical mathematics, Amsterdam (1980).

[8] RHEINBOLDT, W.C., *Methods for solving systems of nonlinear equations,* Conference board of the mathematical sciences 14; Regional conference series in applied mathematics. Published by SIAM Philadelphia P.A.

[9] WACKER H., *Continuation methods,* Proceedings of a Symposium at the University of Linz, Austria, October 3-4, 1977.