A NEW FAST SOLVER——MONOTONE MG METHOD (MMG)*

ZOU JUN (邹 军)
(Wuhan University Wuhan, China)

§0. Introduction

In this paper, we discuss a new method——the MMG for solving a class of linear or nonlinear elliptic boundary value problems, fixed point problems and variational inequality problems. The method is based on the FAS introduced by Brandt[1] and uses nonlinear monotone relaxation iteration for its smoothing part. The difference between the FAS and the MMG methods is an additional parameter d_k to guarantee the monotonicity of the iterative sequence. It is just this parameter that may effectively accelerate the convergence of the FAS for a class of problems discussed here. Its convergence (including v-cycle and w-cycle) can be easily proved and the assumptions are very natural. Numerical experiments and comparisons with the MG and the nonlinear monotone relaxation method are reported.

§1. Problem and Algorithm

Consider the discrete system of equations

$$Lu=f, \tag{1}$$

where $f=(f_1, \cdots, f_n)^T$, $u=(u_1, \cdots, u_n)^T$, $Lu=(L^1u, \cdots, L^nu)^T$.

Suppose L is an M-matrix (if (1) is linear) or M-function (if (1) is nonlinear). The system (1) arises in elliptic boundary value problems, fixed point problems, etc. The iteration method usually used to solve (1) is the SOR iteration, that is,

$$\begin{cases} \text{from } L^i(u_1^{k+1}, \cdots, u_{i-1}^{k+1}, u_i, u_{i+1}^k, \cdots, u_n^k)=f_i, \\ \text{we get } u_i; \text{ then set} \\ u_i^{k+1}=u_i^k+r(u_i-u_i^k),\ r\in(0, 2),\ i=1, \cdots, n,\ k=0, 1, \cdots. \end{cases} \tag{2}$$

We know[8] that if there exist two vectors u^0, v^0 such that

$$u^0\leqslant v^0, \quad Lu^0\leqslant f\leqslant Lv^0,$$

where $u^0\leqslant v^0$ ($u^0<v^0$ below) is defined componently, then for $r\in(0, 1]$, the two sequences $\{u^k\}$, $\{v^k\}$ produced by (2) taking u^0, v^0 as their initial vectors respectively satisfy

$$u^n\uparrow \bar{u},\ v^n\downarrow \bar{v}, \text{ as } n\to\infty,$$

and

$$\bar{u}=\bar{v},\ L\bar{u}=L\bar{v}=f.$$

* Received November 13, 1985.

We have the same result[3] for the Jacobi iteration.

For convenience we denote the monotone iteration ($r\in(0, 1]$) by SUR (successive underrelaxation) and the iterative process by

$$u^{k+1}=S(u^k, L, f).$$

Now we describe our algorithm based on the FAS.

Suppose we have $N+1$ numbers: $h_0>h_1>\cdots>h_N$, and the corresponding grid spaces: $\Omega_0\subset\Omega_1\subset\cdots\subset\Omega_N$ and discrete systems of equations:

$$L_k u^k=f_k, \quad k=0, 1, \cdots, N.$$

Our purpose is to solve the equations

$$L_N u^N=f_N, \quad N\geqslant 1.$$

In the following we use $u^{N,k,t}$ as an iterative vector; the superscripts N, k, t are self-explanatory. I_k^{k+1} and I_{k+1}^k are operators which trans form grid functions on Ω_k into grid functions on Ω_{k+1} and vice versa[2]; we call them prolongator and restrictor, respectively.

Algorithm 1 (MMG).

Starting with a given k-th iterative approximation $u^{N,k,0}$ to u^N:

$$L_N u^{N,k,0}<f_N\,(L_N u^{N,k,0}>f_N).$$

Step 1. Pre-smoothing:

$$u^{N,k,t}=S(u^{N,k,t-1}, L_N, f_N), \quad t=1, 2,\cdots, t_1.$$

Step 2. Coarse-grid correction:

——Compute the defect $\bar{d}_N=f_N-L_N u^{N,k,t_1}$,

——Restrict the defect $\bar{d}_{N-1}=I_N^{N-1}\bar{d}_N$,

——Solve on the N-th grid Ω_{N-1}:

$$L_{N-1}w^{N-1}=L_{N-1}(I_N^{N-1}u^{N,k,t_1})+\bar{d}_{N-1}. \tag{3}$$

If $N=1$, we solve (3) directly.

If $N>1$, we solve (3) by performing $m\geqslant 1$ steps of the MMG N-grid method (using the grids Ω_0, Ω_1, $\cdots$, Ω_{N-1} and the corresponding grid operators) to (3) with $I_N^{N-1}u^{N,k,t_1}$ as first approximation. Denote the approximate solution by $\bar{w}^{N-1}$.

——Compute $\hat{u}^{N,k,t_1}=u^{N,k,t_1}+d_k I_{N-1}^N(\bar{w}^{N-1}-I_N^{N-1}u^{N,k,t_1})$,

where d_k is chosen in such a way that

$$L_N\hat{u}^{N,k,t_1}<f_N\,(L_N\hat{u}^{N,k,t_1}>f_N).$$

Step 3. Post-smoothing:

$$u^{N,k,t_1+1}=S(\hat{u}^{N,k,t_1}, L_N, f_N),$$

$$u^{N,k,t}=S(u^{N,k,t-1}, L_N, f_N),\ t=t_1+2, \cdots, t_1+t_2.$$

Continue the above process with $k+1$ instead of k and $u^{N,k+1,0}=u^{N,k,t_1+t_2}$.

Remark 1. Clearly d_k in Algorithm 1 exists ($d_k>0$), and usually $d_k\geqslant 1$. If we set $d_k\equiv 1$, then Algorithm 1 reduces to the FAS. For concrete problems, d_k can be easily computed.

§ 2. Convergence of Algorithm 1

First, we generalize a result from [3]. It is a basic conclusion for this paper.

Lemma 1. *Let F: $D\subset R^N\to R^N$ be a continuous, off-diagonally antitone and strictly diagonally isotone mapping. And assume x^0, $y^0\in D$, $b\in R^N$ satisfy*

$$x^0<y^0,\quad Fx^0<b<Fy^0.$$

Then, for $r\in(0, 1)$, the iterates $\{x^k\}$, $\{y^k\}$ produced by the SUR satisfy

$$x^0<x^k<x^{k+1}<y^{k+1}+y^k<y^0,\quad Fx^k<b<Fy^k,\ k=1,\ 2,\ \cdots$$

and

$$x^k\uparrow\bar{x},\quad y^k\downarrow\bar{y},\quad F\bar{x}=F\bar{y}=b.$$

The proof of the lemma is the same as in [3], with only slight modifications.

Next, we discuss the convergence of Algorithm 1 in the case of two grids. We have

Theorem 1. *Suppose L_1, L_0 are continuous M-functions on their domains of definition Ω_1 and Ω_0, respectively. And there are vectors $u^{1,0,0}$, $v^{1,0,0}$ and f_1 such that*

$$u^{1,0,0}<v^{1,0,0},\quad L_1u^{1,0,0}<f_1<L_1v^{1,0,0}.$$

Also assume that the prolongator I_0^1 and restrictor I_1^0 in Algorithm 1 are linear strictly monotone operators (that is, $u>v\Rightarrow Ku>Kv$, K is I_1^0 or I_0^1). And $t_1\geqslant1$, $t_2\geqslant0$ (or $t_1\geqslant0$, $t_2\geqslant1$). Then the sequence given by Algorithm 1 taking $u^{1,0,0}$ as its initial vector is strictly monotone increasing and converges to the exact solution u^1 of $L_1u^1=f_1$.

We have the same result for $v^{1,0,0}$, but the corresponding sequence is strictly monotone decreasing.

Remark 2. The restrictions of prolongator I_0^1 and restrictor I_1^0 are easily satisfied for the usual operators

Proof. From Lemma 1, we know

$$\begin{gathered}u^{l,k,t-1}<u^{l,k,t}<v^{l,k,t}<v^{l,k,t-1},\\ L_1u^{l,k,t}<f_1<L_1v^{l,k,t},\quad t=1,\ 2,\ \cdots,\ t_1.\end{gathered}\tag{4}$$

This fact, with the assumption on I_1^0, shows

$$\bar{d}_0=I_1^0\bar{d}_1=I_1^0(f_1-L_1u^{l,k,t_1})>0.$$

Therefore

$$L_0w^0=L_0(I_1^0u^{l,k,t_1})+\bar{d}_0>L_0(I_1^0u^{l,k,t_1}).$$

Owing to the assumption on L_0, we have

$$w^0>I_1^0u^{l,k,t_1},$$

which implies

$$\hat{u}^{l,k,t_1}=u^{1,k,t_1}+d_kI_0^1(w^0-I_1^0u^{l,k,t_1})>u^{l,k,t_1}.$$

From the definition of d_k, we obtain

$$L_1\hat{u}^{l,k,t_1}<f_1.\tag{5}$$

Similarly, we have

$$\begin{gathered}v^{l,k,t_1}>\hat{v}^{l,k,t_1},\\ L_1\hat{v}^{l,k,t_1}>f_1.\end{gathered}\tag{6}$$

(5), (6) and the assumption on L_1 show

$$\hat{u}^{l,k,t_1} < \hat{v}^{l,k,t_1}.$$

Again using Lemma 1 and (4), we get

$$u^{l,0,0} < u^{l,k,t-1} < u^{l,k,t} < v^{l,k,t} < v^{l,k,t-1} < v^{1,0,0},$$
$$L_1 u^{l,k,t} < f_1 < L_1 v^{k,k,t} \quad t=1,\cdots,t_1+t_2. \tag{7}$$

So, there exist two vectors $\bar{u}$, $\bar{v}$ such that

$$u^{l,k,t} \to \bar{u},\ v^{l,k,t} \to \bar{v},\ \text{as } k\to\infty,$$
$$\bar{u} \leqslant \bar{v},\ t=0,\ 1,\cdots,t_1+t_2. \tag{8}$$

Now we show $\bar{u}=\bar{v}$ and $L_1\bar{u}=f_1$. Denote L_1 by $L_1=(L_1^1,\cdots,L_1^N)$

Consider two real continuous functions:

$$x(s) = L_1^i(u_1^{t_1},\cdots,u_{i-1}^{t_1},\ s,\ u_{i+1}^{t_1-1},\cdots,\ u_N^{t_1-1})$$
$$y(s) = L_1^i(v_1^{t_1},\ \cdots,\ v_{i-1}^{t_1},\ s,\ v_{i+1}^{t_1-1},\cdots,\ v_N^{t_1-1}),\quad 1\leqslant i\leqslant N.$$

Note that we omit the superscripts 1, k. Clearly

$$x(s) \geqslant y(s),\quad \forall s\in R^1.$$

This shows, with (7),

$$y(u_i^{t_1-1}) \leqslant x(u_i^{t_1-1}) \leqslant L_1^i(u^{t_1-1}) < f_1^i < L_1^i(v^{t_1-1})$$
$$\leqslant y(v_i^{t_1-1}) \leqslant x(v_i^{t_1-1}).$$

By the assumption on L_1, there exist unique $\bar{u}_i^{t_1-1}$ and $\bar{v}_i^{t_1-1}$ which satisfy

$$u_i^{t_1-1} < \bar{u}_i^{t_1-1} \leqslant \bar{v}_i^{t_1-1} < v_i^{t_1-1},$$

and

$$x(\bar{u}_i^{t_1-1}) = f_1^i = y(\bar{v}_i^{t_1-1}).$$

Recalling the roles of $\bar{u}_i^{t_1-1}$ and $\bar{v}_i^{t_1-1}$ in Algorithm 1, we know

$$u_i^{t_1} = u_i^{t_1-1} + r(\bar{u}_i^{t_1-1} - u_i^{t_1-1}),$$

that is, $\bar{u}_i^{t_1-1} = (u_i^{t_1} - u_i^{t_1-1})/r + u_i^{t_1-1} \to \bar{u}_i$, as $k\to\infty$.

Similarly, we have

$$\bar{v}_i^{t_1-1} \to \bar{v}_i,\ \text{as } k\to\infty,\quad 1\leqslant i\leqslant N.$$

Set $s=\bar{u}_i^{t_1-1}$ and $\bar{v}_i^{t_1-1}$ in $x(s)$ and $y(s)$, respectively. Then as $k\to\infty$, we obtain (with (8))

$$L_1^i\bar{u} = f_1^i = L_1^i\bar{v},\quad 1\leqslant i\leqslant N,$$

which show $\bar{u}=\bar{v}$. So we complete the proof of Theorem 1.

We can easily generalize Theorem 1 (two-grid iteration) to the multigrid iteration by induction.

Suppose we have $N+1$ numbers: $h_0>h_1>\cdots>h_N$. Our purpose is to solve $L_N u^N = f_N$. We will use grid operators $L_k(k=0,\ 1,\ \cdots,\ N-1)$. We have

Theorem 2. *Suppose $L_i (i=0,\ 1,\ \cdots,\ N)$ are continous M-functions on Ω_i, respectively, and there are vectors f_N, $u^{N,0,0}$ and $v^{N,0,0}$ which satisfy*

$$L_N u^{N,0,0} < f_N < L_N v^{N,0,0}.$$

The restrictions of prolongator I_{k-1}^k and restrictor I_k^{k-1} are the same as I_0^1 and I_1^0 in Theorem 1. Then the sequence given by Algorithm 1 taking $u^{N,0,0}$ as its initial vector is *strictly monotone increasing and converges to the exact solution of* $L_N u^N = f_N$.

We have the same result for $v^{N,0,0}$, but the corresponding sequence is strictly monotone decreasing.

§ 3. Algorithm and Convergence in the Case of Non-Strict Monotonicity

In this section, we only assume that the initial vectors u^0 and v^0 satisfy

$$Lu^0 \leqslant f \leqslant Lv^0.$$

All the signs in this section are the same as in Sections 1, 2.

Our purpose is to solve $L_N u^N = f_N$, $N \geqslant 1$. In the following we use the notation $I(u^N) = \{i: L_N^i u^N = f_N^i\}$ where i means the i-th component of the corresponding vector.

Algorithm 2.

Starting with a given k-th iterative approximation $u^{N,k,0}$ to u^N:

$$L_N u^{N,k,0} \leqslant (\geqslant) f_N.$$

Step 1. As step 1 in Algorithm 1.

Step 2. As step 2 in Algorithm 1, but change the computing of $\hat{u}^{N,k,t_1}$. Denote $I_{N-1}^N(\bar{w}^{N-1} - I_N^{N-1} u^{N,k,t_1})$ by e_N.

——Compute $\hat{u}^{N,k,t_1} = u^{N,k,t_1} + d_k \bar{e}_N$ where $d_k > 0$ satisfies:

$$L_N^i \hat{u}^{N,k,t_1} < (>) f_N^i, \text{ for all } i: L_N^i u^{N,k,t_1} < (>) f_N^i,$$

$$\bar{e}_N^i = \begin{cases} e_N^i, \text{ for all } i: f_N^i > (<) L_N^i u^{N,k,t_1}, \\ 0, \text{ for } i \in I(u^{N,k,t_1}). \end{cases}$$

Step 3. As step 3 in Algorithm 1.

We have the following

Theorem 3. *Take the assumptions of Theorem 2 and assume the grid functions L_i on Ω_i:*

$$L_i u = A_i u + B_i u, \quad i = 0, 1, \cdots, J,$$

where A_i are $N_i \times N_i$ M-matrices, B_i: $R^{N_i} \to R^{N_i}$ are diagonal, continuous and isotone functions. Then the conclusions of Theorem 2 hold, but the corresponding iterative sequens are monotone, not strictly monotone.

Proof. We only need to prove the following results:

Suppose L, $A = (a_{ij})$ and B have the same properties as L_i, A_i, B_i. And assume that there are vectors u, $w \in R^N$ such that

(a) $$Lu \leqslant f, \quad w \geqslant 0,$$

where $w_i = 0$, for $i \in I(u) = \{i: f^i = L^i u\}$. Then, if $d > 0$ satisfies

$$L^i(u + dw) < f^i, \text{ for } i: f^i > L^i(u),$$

we have $L(u + dw) \leqslant f$;

or such that

(b) $$Lu \geqslant f, \quad w \leqslant 0,$$

where $w_i = 0$, for $i \in I(u) = \{i: f^i = L^i u\}$. Then, if $d > 0$ satisfies

$$L^i(u+dw)>f^i, \text{ for } i: f^i<L^iu,$$

we have $L(u+dw)\geqslant f$.

The proof of (a): Consider the jth component of $L(u+dw)$, $j\in I(u)$

$$L^j(u+dw)=A^j(u+dw)+B^j(u_j+dw_j)=A^ju+dA^jw+B^j(u_j)$$
$$=L^ju+dA^jw=f^j+d\sum_{k\neq j}a_{jk}w_k\leqslant f^j, \quad (\text{with } a_{jk}\leqslant 0,\ j\neq k)$$

Also, $L^j(u+dw)<f^j$, for j: $L^ju<f^j$, that is, $L(u+dw)\leqslant f$.

Similarly, we can prove (b).

With (a) and (b), the proof of Theorem 3 can proceed in the same way as Theorems 1, 2.

§ 4. A concrete Problem

In this section, we consider a mildly nonlinear partial differential equation of elliptic type

$$\begin{aligned} &\Delta u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=f(u), \quad \text{in } \Omega, \\ &u|_{\partial\Omega}=g(x, y), \end{aligned} \tag{9}$$

where $\frac{df}{du}\geqslant 0$.

Using the standard five-point difference scheme with lexicographic ordering of the grid points (from left to right and from bottom to top), discretize (9) (step size h):

$$L_hu=A_hu+h^2B_hu=f_h \tag{10}$$

where $u=(u_1, \cdots, u_N)^T$, $f_h=(f_1, \cdots, f_N)^T$, $B_hu=(f(u_1), \cdots, f(u_N))^T$,

$$A_h=\begin{pmatrix} C & -I & & & \\ -C & C & -I & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & -I & C & -I \\ & & & -I & C \end{pmatrix}, \quad C=\begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

Clearly, L_h is an M-matrix (if B_hu is linear on u) or M-function (if B_hu is nonlinear on u).

Remark 3. Obviously, if the coefficients of $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$ are nonnegative functions of x and y and at least one of them is positive or $f(u)$ is replaced by $f(x, y, u)$, we can obtain the same result.

Remark 4. In the 1-D case, we have a similar discrete structure.

For other kinds of examples, refer to [6].

§ 5. Numerical Experiments

Now we give some numerical results for solving the following two problems:

(A) $$\Delta u=e^u+f(x, y),$$

(B) $\Delta u=u^3-g(x, y)$,

where $f=4-e^{x^2+y^2+1}$, $g=(x^2+y)^3-2$, $\Omega=(0, 1)^2$.

Their exact solutions are $u=x^2+y^2+1$ and $u=x^2+y$, respectively. Discretizing problems (A), (B) in the same way as in Section 4, we obtain

$$L_h u=A_h u+h^2 B_h u=f_h,$$

where A_h, f_h, u_h are as in Section 4, $B_h u=(e^{u_1}, \cdots, e^{u_N})^T$ for problem (A′), $B_h u=(u_1^3, u_N^3)^T$ for problem (B).

For the sake of comparison with the SUR and the MG methods (i.e. $d_k=1$ in Algorithm 1), we use $\text{EPS}=\max_{1<i<N}|u_i^k-\bar{u}_i|$ as the error control, where u^k is the k-th iterative solution and $\bar{u}$ is the exact solution of problem (A) or (B). The time and number of iterations taken by the methods above are listed when EPS reaches a certain accuracy.

It is well known that if the number of grids is too small, the advantages of the MG method cannot show up. Here we only give the results of two cases i.e. four grids $\left(h=\frac{1}{24}, \frac{1}{12}, \frac{1}{6}, \frac{1}{3}\right)$ and five grids $\left(h=\frac{1}{48}, \frac{1}{24}, \frac{1}{12}, \frac{1}{6}, \frac{1}{3}\right)$.

Some notations. In the following tables, $x_0<0$ means $x_0=-A_h^{-1}(h^2B_h0-f_h|+h^2e)$: $L_hx_0<f_h$; $x_0>0$ means $x_0=A_h^{-1}(h^2B_h0-f_h|+h^2e)$: $L_hx_0>f_h$, where $e=(1, 1, \cdots, 1)^T$. For problem (A), we also consider the case $x_0=0$ which does not satisfy $L_hx_0<f_h$, but approximately so. For the casses $x_0<0$ and $x_0=0$, the iterative sequences are strictly monotone increasing, while for $x_0>0$, strictly monotone decreasing.

We take the relaxation factors $\tau=1.0$ for the SUR, 0.98 or 0.9 for the MG and the MMG, respectively. The rate of convergence of the SUR is faster in the case $\tau=1.0$ than in the case $\tau<1.0$. Although we only take $\tau=0.98$ for the MG and the MMG, their rates of convergence are much greater than the SUR in the case $\tau=1.0$.

All the numerical results reported here were computed on an M-340S system made in Japan.

We take $t_1=2$, $t_2=1$ in Algorithm 1. Time in the following tables means CPU time (M-minute, S-second), ITER means the number of iterations, and MG-V (also MG-W, MMG-V) means V-cycle or W-cycle.

Table 1 For problem (B): EPS$=10^{-4}$, I_{h+1}^k-FW, I_k^{k+1}-linear interpolation [2], Four-grid: $h_0=\frac{1}{3}$, $h_1=\frac{1}{6}$, $h_2=\frac{1}{12}$, $h_3=\frac{1}{24}$

Method	x_0	τ	Time	Iter	τ	Time	Iter
SUR			1M 11S	647	1.0	1M 2S	534
MG-V	$x_0<0$	0.9	50S	88	0.98	47S	81
MG-W			43S	57		41S	53
MMG-V (2, 5)			19S	14		19S	12
MMG-W (2, 5)			18S	9		16S	6
SUR			54S	510	1.0	52S	479
MG-V	$x_0>0$	0.9	38S	65	0.98	37S	61
MG-W			34S	41		32S	39
MMG-V (2, 5)			17S	9		17S	9
MMG-W (2, 5)			15S	4		14S	4

Table 2 For problem (A): EPS $=10^{-3}$, $I_{k+1}^{k}-FW$, I_k^{k+1} as in Table 1, step size as in Table 1

Method		x_0	r	Time	Iter	r	Time	Iter
SUR		$x_0<0$	0.9	1M 21S	451	1.0	1M 16S	421
MG-V		$x_0<0$	0.9	54S	61	0.98	53S	59
MG-W				47S	40		46S	38
MMG-V	(2, 5)			20S	9		20S	9
MMG-W	(2, 5)			17S	4		17S	4
SUR		$x_0>0$	0.9	56S	304	1.0	53S	291
MG-V				42S	42	0.98	40S	40
MG-W				38S	28		35S	27
MMG-V	(2, 5)			17S	6		17S	6
MMG-W	(2, 5)			15S	3		15S	3
SUR		$x_0=0$	0.9	1M 11S	409	1.0	1M 1S	379
MG-V				48S	56	0.98	47S	53
MG-W				42S	37		42S	36
MMG-V	(2, 5)			17S	8		17S	8
MMG-W	(2, 5)			15S	4		15S	4

Table 3 For problem (B): $I_{k+1}^{k}-INJ$ [2] ($r=0.98$), $I_{k+1}^{k}-FW$ ($r=0.9$), I_k^{k+1} as in Table 1, EPS $=10^{-3}$. Five-grid: $h_0=\frac{1}{3}$, $h_1=\frac{1}{6}$, $h_2=\frac{1}{12}$, $h_3=\frac{1}{24}$, $h_4=\frac{1}{48}$

Method		x_0	r	Time	Iter	r	Time	Iter
SUR		$x_0<0$	0.9	15M 42S		1.0	14M 27S	
MG-V				6M 20S	185	0.98	6M 4S	171
MG-W				4M 7S	81		3M 47S	76
MMG-V	(2, 5)			1M 21S	20		1M 16S	18
MMG-W	(2, 5)			50S	5		51S	5
SUR		$x_0>0$	0.9	10M 55S		1.0	10M 3S	
MG-V				4M 36S	140	0.98	4M 11S	126
MG-W				2M 55S	60		2M 45S	56
MM-V	(2, 5)			1M 5S	14		1M 2S	13
MMG-W	(2, 5)			47S	4		46S	4

Table 4 For problem (A): I_k^{k+1}, I_{k+1}^{k}, EPS and step size as in Table 3

Method		x_0	r	Time	Iter	r	Time	Iter
SUR		$x_0<0$	0.9	21M 11S		1.0	17M 49S	
MG-V				10M 10S	225	0.98	9M 42S	193
MG-W				6M 39S	103		5M 45S	88
MMG-V	(2, 5)			1M 7S	9		1M	7
MMG-W	(3, 5)			1M 1S	5		1M 6S	6
SUR		$x_0>0$	0.9	11M 15S		1.0	11M 3S	
MG-V				5M 27S	118	0.98	5M 10S	110
MG-W				3M 42S	54		3M 28S	51
MMG-V	(3, 5)			50S	5		50S	5
MMG-W	(3, 5)			57S	5		52S	4
SUR		$x_0=0$	0.9	18M 40S		1.0	15M 22S	
MG-V				8M 57S	209	0.98	7M 46S	176
MG-W				5M 48S	96		4M 54S	81
MMG-V	(3, 5)			40S	8		87S	7
MMG-W	(3, 5)			38S	5		37S	5

Remark 5. The parameter d_k in Algorithm 1 is chosen as follows: Suppose vectors U, $v \in R^N$ satisfy $V>0$, $L_h U<f_h$. Then take

$$d=\min\left\{\min_{z_h^i>0}\frac{f_h^i-L_h^i U}{z_h^i},\ d_0\right\},$$

where $d_0>0$ (a given constant) and $z_h^i=L_h^i V+h^2(v_i e^{u_i+d_0 v_i}-e^{v_i})$ for problem (A), and $L_h^i V+h^2(3u_i^2 v_i+3d_0|u_i|v_i^2+(d_0^2-1)v_i^3)$ for problem (B). In the other case, i.e. $L_h U>f_h$, take $d=\min\left\{\min_{z_i<0}\frac{f_h^i-L_h^i U}{z_i},\ d_0\right\}$, where $z_i=L_h^i V+h^2(3u_i^2 v_i-3d_0|u_i|v_i^2+(d_0^2-1)v_i^3)$ for problem (B), and $\min_{z_i<0}\frac{f_h^i-L_h^i U}{z_i}$ for problem (A).

In the tables above, the sign (p, q) under MMG-V or MMG-W means taking $d_0=p$ on the lower grids, and $d_0=q$ on the highest grid.

From the choice os d_k above we know that d_0 cannot be taken too large; otherwise d_k may be very small (then we have a slower rafe of convergence). Usually we take $d_0\in[1, 4]$ on the lower grids, and $d_0\in[5, 6]$ on the highest grid. It is better to take $d_0\in[2, 3]$ on the lower grids, and $d_0\approx 5$ on the highest grid. But the best choice of d_0 still needs further analysis and more numerical experiments.

Remark 6. The numerical experiments show that the MMG method is more efficient than the MG method if we choose d_0 appropriately. The number of iterations of the MMG clearly shows its fast convergence.

Finally we list the errors EPS after each iteration for the MMG method in three cases: (for problem (A))

Table 5 MMG-W: $x_0=0$, $r=0.98$, I_{k+1}^k-FW, I_k^{k+1} as above, five-grid, step size as Table 3 (3, 5)

number of iterations	0	1	2	3	4	5
EPS	.2918E01	.6089E00	.1497E00	.1419E-1	.1436E-2	.9565E-3

Table 6 MMG-W: $x_0=0$, I_{k+1}^k, I_k^{k+1} and r as Table 5, four-grid (2, 5)

number of iterations	0	1	2	3	4
EPS	.2837E01	.4528E00	.4178E-1	.4174E-2	4063E-3

Table 7 MMG-W: $x_0<0$, others as in Table 6 (2, 5)

number of iterations	0	1	2	3	4
EPS	.477E01	.8893E00	.7693E-1	.8096E-2	.6742E-3

§ 6. Numerical Results of Some Typical M-Type or Non-M Type Problems

For many typical M-type or non-M type problems, namely the coefficient functions of discrete systems are not M-functions, we have made a lot of numerical experiments. The results show that the initial iteration vector for the MMG method is not necessarily chosen strictly to satisfy $Lu^0 \leqslant f$ or $Lu^0 \geqslant f$ as Algorithm 1 or 2, and the MMG method can effectively be used for more problems.

The following results are computed by using the five-grid MMG iteration, namely $h_0 = \frac{1}{3}$, $h_1 = \frac{1}{6}$, $h_2 = \frac{1}{12}$, $h_3 = \frac{1}{24}$, $h_4 = \frac{1}{48}$, and $\gamma = 0.98$, EPS$=10^{-4}$, $x_0 = 0$. We discretize the following problems a), b) and c).

a) Chemical equilibrium problem

$$\begin{cases} \Delta u = u^2, & \text{in } \Omega = (0, 1)^2, \\ u = g(x, y), & \text{on } \Gamma, \end{cases}$$

where $g(x, y) \leqslant 0$. We know that there exists a nonnegative unique solution for the problem[4]. We take $g(x, y) = x^2 + y^2 + 1$. The numerical results are as follows:

method	MG-W	MG-V	MMG-W (2, 5)	MMG-V (2, 5)
Iter	75	149	8	21
Time	3M 27S	4M 57S	35S	58S

Especially, we have got the same results by using $x_0 = -\frac{1}{4}(h^2 e + f_h) < 0$.

b) Consider the following problem:

$$\begin{cases} \Delta u = u^2, & \text{in } \Omega = (0, 1), \\ u = \varphi(x, y), & \text{on } \Gamma, \end{cases}$$

where $\varphi(x, y) \leqslant 0$, and $\varphi(x, y) \not\equiv 0$.

Whether there exists a solution for the problem has not been proved yet. But using the MMG method, we have made a lot of numerical experiments for different $\varphi(x, y)$, and the results show that the iterative sequence for the MMG converges to the exact solution of the discrete system when we take $\varphi(x, y) \geqslant -4.5$. When $\varphi(x, y) \leqslant -5$, the iterative sequence does not converge; perhaps it is because there does not exist any solution for the problem in this case. The following are two numerical results by using the MMG-W method:

(3, 5)

$\varphi(x, y)$	Iter	Time
$-3.0+\sin(xy)$	8	34.72S
-4.5 (for four-grid)	7	15.065

c) A simple bifurcation problem

$$\begin{cases} \Delta u = -\lambda e^u, & \text{in } \Omega = (0, 1)^2, \\ u = 0, & \text{on } \Gamma. \end{cases}$$

We know an upper bound of the bifurcation point is 6.81. The following are some numerical results by using MMG-W:

(4, 5)

value	Iter	Time	$u_h\left(\frac{1}{2}, \frac{1}{2}\right)$	EPS
1	5	32.88S	0.0780582	10^{-4}
6.78	6	38.78S	1.12644	10^{-8}

6.8	8	42.28S	1.13184	10^{-3}
6.8099	19	1M31S	1.138987	10^{-3}

When $\lambda \geqslant 6.81$, the error EPS does not reduce, and the solution of the bifurcation problem does not exist in the case. The above result for $\lambda=1$ is the same as that in [5].

Acknowledgements. The author would like to thank Profs. Maoxiang Li and Lishan Kang for their instructions and helpful discussions.

The author also owes sincere thanks to Prof. Wolfgang Hackbusch for reading the early drafts of the manuscript and for pleasant discussions concerning the method in this paper during his visit in China (1985).

References

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

[2] Lecture Notes in Mathemasics, 960, Multigrid Methods (W. Hackbusch, U. Trottenberg, eds.), 1—219.

[3] J. M. Ortega, W. Rheinboldt, Iterative Solutions of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[4] C. M. Ablow, C. L. Perry, Iterative solutions of the Dirichlet Problem for $\Delta u=u^2$, *SIAM J. Appl. Math.*, **7** (1959), 459—467.

[5] W. Hackbusch, On the fast solutions of nonlinear elliptic equations, *Numer. Math.*, **32** (1979), 83—95.

[6] Zou Jun, A new fast solver-monotone multigrid method, M D. thesis, Wuhan University, 1986.