Vol. 5 No. 4 JOURNAL OF COMPUTATIONAL MATHEMATICS October 1987

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[3] 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 $\Omega_0,\ \Omega_1,\ \cdots,\ \Omega_{N-1}$ 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.