Vol. 4 No. 2 JOURNAL OF COMPUTATIONAL MATHEMATICS April 1986

MULTIGRID METHOD FOR ELASTICITY PROBLEMS*

WANG JIN-XIAN (王烇賢) HUANG YU-XIA (黄玉霞)
GONG JIA-YAO (龚家跃)
(Computing Center, Academia Sinica, Beijing, China)

§ 1. Introduction

The multigrid method is a fast iterative method developed during the sixties for solving elliptic partial differential equations with boundary value condition. Its difference from other iteration methods is that the convergence rate is independent of the grid size h, and the operating amount for obtaining the approximate solutions with the same accuracy is $O(N)$ where N is the number of the unknowns in equations. Since the seventies the MG method has been widely used in many other problems successfully, including some elliptic variational inequalities, whose physical background includes some fluid flow through porous medium, and the water cone problem of an oil well[3,8,9]. This paper discusses how the MG method is used in some elastic mechanical problems, including general two- or three-dimensional elasticity problems and a type of variational inequality of elasticity problems, whose physical background is elastic-rigid and elastic-elastic contact problems without friction. We will prove the convergence of the method and give some numerical examples.

We would like to thank Zhou Shu-zi for his helpful discussion and suggestion.

§ 2. Mathematical Problems and MG Method

Consider an elastic body Ω and a rigid body Ω_1 and suppose that $x=(x_1, x_2, x_3)$ is a set of Cartesian coordinates of a point in the space, $u(x)=(u_1(x), u_2(x), u_3(x))$ is its displacement function, ε_{ij}, σ_{ij} are tensors of strain and stress respectively, $f=(f_1, f_2, f_3)$ is a vector of the body force, $p=(p_1, p_2, p_3)$ is a vector of the surface force, the boundary of Ω, $\partial\Omega=\Gamma_u\cup\Gamma_\sigma\cup\Gamma_c$. On Γ_u and Γ_σ displacements and surface forces are prescribed respectively. Γ_c is the boundary where Ω and Ω_1 are contacting or will contact. So Γ_c is unknown in the problem and is assumed to be a smooth surface. $n=(n_1, n_2, n_3)$ is a unit external normal vector of $\partial\Omega$. Now we consider two types of problems: (I) linear problems, (II) nonlinear problems.

Problem (I.1). Find a displacement function $u(x)$, satisfying the following relations:

$$\varepsilon_{ij}(u)=\varepsilon_{ji}(u)=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right), \tag{2.1}$$

$$\sigma_{ij}(u)=\sigma_{ji}(u)=\lambda\sum_{k=1}^{3}\varepsilon_{kk}(u)\delta_{ij}+2\mu\varepsilon_{ij}(u), \tag{2.2}$$

* Received March 1, 1985.

$$\sigma_{ij,j}+f_i=0, \quad x\in\Omega,\ i,\ j=1,\ 2,\ 3, \tag{2.3}$$

$$\sigma_{ij}n_j=p_i, \quad x\in\Gamma_\sigma, \tag{2.4}$$

$$u_i=\bar{u}_i, \quad x\in\Gamma_u, \tag{2.5}$$

where δ_{ij} is Kronecker's symbol, and λ, μ are two constants which are assumed to satisfy $\lambda>0$, $\mu>0$. Conventionally, repeated subscript i denotes summation for $i=1,\ 2,\ 3$.

Problem (II.1). Find a displacement vector $u(x)$ and the contact boundary Γ_c, satisfying (2.1)—(2.5) and relations:

$$\left.\begin{aligned} &u_in_i-s\leqslant 0,\ \sigma_n\leqslant 0\\ &\sigma_n\cdot(u_in_i-s)=0\end{aligned}\right\}\ \text{on } \Gamma_c, \tag{2.6}$$

where s is the initial gap between Ω and Ω_1, σ_n is normal stress

$$\sigma_n=\sigma_{ij}n_in_j. \tag{2.7}$$

We define the function space H and its convex subset K

$$H=\{u=(u_1,\ u_2,\ u_3)\,|\,u_i\in H^1(\Omega),\ u_i=\bar{u}_i,\ x\in\Gamma_u\}, \tag{2.8}$$

$$K=\{u\in H,\ u_in_i-s\leqslant 0,\ x\in\Gamma_c\}. \tag{2.9}$$

The norm in H is

$$\|v\|_1=\left\{\int_\Omega (v_iv_i+v_{i,j}v_{i,j})dx\right\}^{1/2}. \tag{2.10}$$

Define a bi-linear functional and a linear functional as follows:

$$a(u,\ v)=\int_\Omega \sigma_{ij}(u)\varepsilon_{ij}(v)dx, \tag{2.11}$$

$$f(v)=\int_\Omega f_iv_i\,dx+\int_{\Gamma_\sigma} p_iv_i\,ds. \tag{2.12}$$

According to the Korn inequality, $a(u,\ v)$ is H-coercive, i.e. there exists $\alpha>0$ such that

$$a(u,\ v)\geqslant\alpha\|v\|_1^2, \quad \forall v\in H. \tag{2.13}$$

Problems (I.1), (II.1) have the following equivalent variational problems[6,7]:

Problem (I.2).

$$\begin{cases}\text{Find } u\in H \quad \text{such that}\\ a(u,\ v)=f(v), \quad \forall v\in H.\end{cases} \tag{2.14}$$

Problem (I.3).

$$\begin{cases}\text{Find } u\in H \quad \text{such that}\\ J(u)\leqslant J(v), \quad \forall v\in H,\end{cases} \tag{2.15}$$

where $J(v)=\frac{1}{2}a(v,\ v)-f(v)$.

Problem (II.2).

$$\begin{cases}\text{Find } u\in K \quad \text{such that}\\ a(u,\ v-u)\geqslant f(v-u), \quad \forall v\in K.\end{cases} \tag{2.16}$$

Problem (II.3).

$$\begin{cases}\text{Find } u\in K \quad \text{such that}\\ J(u)\leqslant J(v), \quad \forall v\in K.\end{cases} \tag{2.17}$$

By the general finite element method the problems above have the following discrete forms respectively:

Problem (I.4).

$$\begin{cases} \text{Find } u\in R^N \quad \text{such that} \\ J_N(u)\leqslant J_N(v), \quad \forall v\in R^N, \end{cases} \tag{2.18}$$

where $J_N(v)=\frac{1}{2}v^TAv-v^TF$, A is a coefficient matrix (stiffness matrix). From (2.12), (2.13), A is symmetrical and positive definite. F is a vector of equivalent node force, $v=(v_1, v_2, \cdots, v_N)$ where $v_k=(v_{k,1}, v_{k,2}, v_{k,3})$.

Problem (II.4).

$$\begin{cases} \text{Find } u\in K^N \quad \text{such that} \\ J_N(u)\leqslant J_N(v), \quad \forall v\in K^N, \end{cases} \tag{2.19}$$

where $K^N=\{v=(v_1, v_2, \cdots, v_N)\,|\,v_k=(v_{k,1}, v_{k,2}, v_{k,3}),\ (v_{k,i})n_i-s\leqslant 0 \text{ on } \Gamma_c\}$.

We introduce a local coordinate system (τ_1, τ_2, τ_3), where τ_3 is the unit outward normal on Γ_c, and τ_1, τ_2 are tangent vectors. (τ_1, τ_2, τ_3) forms a right hand system.

In this system, the displacement vector $v=(v_1, v_2, v_3)$ is changed to $\tilde{v}=(\tilde{v}_1, \tilde{v}_2, \tilde{v}_3)$ where $\tilde{v}_3=v_in_i$, n_i being the direct confine of τ_3. Thus, the restrictive condition $v_in_i-s\leqslant 0$ is changed to $\tilde{v}_3\leqslant s$. At other nodes, which are not on Γ_c, we still use the global system. To simplify the symbols, we retain the original symbols A, v, F. We denote I_c as the numbers of the normal components of the nodes on Γ_c, i.e.

$$I_c=\{3i_1, 3i_2, \cdots, 3i_k\}, \quad p_{i_k}\in\Gamma_c.$$

So K^N can be written as

$$K^N=\{v=(v_1, \cdots, v_N)\,|\,v_{k,3}\leqslant s_k,\ 3k\in I_c\}. \tag{2.20}$$

Apparently K^N is a convex subset in R^N.

Let

$$v_c=(v_{i_1,3}, v_{i_2,3}, \cdots, v_{i_k,3}), \qquad 3i_k\in I_c. \tag{2.21}$$

From the Kuhn–Tucker theorem[10], problem (II.4) has the following equivalent form, the so called linear complementary problem, whose solution is unique:

Problem (II.5).

$$\begin{cases} \text{Find } U\in K^N \quad \text{such that} \\ AU-F\geqslant 0, \\ U_c\geqslant S, \\ U_c^T(AU-F)_c=0, \\ U|_{\Gamma_u}=\bar{u}, \end{cases} \tag{2.22}$$

where $S=(s_1, s_2, \cdots, s_k)$, $(AU-F)_c=\{(AU-F)_{i_1,3}, \cdots, (AU-F)_{i_k,3}\}^T$, $3i_k\in I_c$.

Problems (II.5), (II.6) can be solved by using the PSOR method[4,7]

$$\begin{cases} \tilde{u}_i^{(k+1)}=u_i^{(k)}+\dfrac{\omega}{a_{ii}}\Big(f_i-\sum_{j<i}a_{ij}u_j^{(k+1)}-\sum_{j\geqslant i}a_{ij}u_j^{(k)}\Big), \quad i=1, \cdots, 3N, \\ u_i^{(k+1)}=\begin{cases} \max\{s_i, \tilde{u}_i^{(k+1)}\}, & \text{if } i\in I_c, \\ \tilde{u}_i^{(k+1)}, & \text{others.} \end{cases} \end{cases} \tag{2.23}$$

The algorithm is convergent for $0<\omega<2$.

For linear problem (I), take $u_i^{(k+1)}=\tilde{u}_i^{(k+1)}$, $i=1, 2, \cdots, 3N$. The PSOR method then becomes a general SOR method.

MG Method.

We select a sequence of meshsize $h_0>h_1>\cdots>h_{m-1}>h_m$; the related grids are $G_0\subset G_1\subset\cdots\subset G_{m-1}\subset G_m$. Generally, we let

$$h_l=\frac{1}{2}h_{l-1}=\cdots=\frac{1}{2^l}h_0$$

and N_l is the number of the nodes of grid G_l, U^l is the displacement vector on G_l, U_c^l is a normal component on contact surface Γ_c, I_c^l is the subset of I_c corresponding to grid G_l ($I_c^m=I_c$), and G_0 is the coarsest grid.

We use the FAS scheme[2,3]. On G_m, we solve problem (P_m)

$$\begin{cases} A_mU^m-F^m\geqslant 0, \\ U_c^m\geqslant S^m, \\ (U_c^m-S^m)^T(A_mU^m-F^m)_c=0, \\ U^m=\bar{u} \quad \text{on } G_m\cap\Gamma_u, \end{cases} \tag{2.24}$$

where $A_m=A$, $F^m=F$, $S^m=S$, $U^m=U$.

Define a restrictive operator I_m^{m-1} and an interpolation operator I_{m-1}^m:

$$\begin{cases} I_m^{m-1}: G(\Omega_m)\to G(\Omega_{m-1}), \\ I_{m-1}^m: G(\Omega_{m-1})\to G(\Omega_m), \end{cases} \tag{2.25}$$

where $G(\Omega_m)$ is the linear space of all grid functions on Ω_m. Suppose that u^m is an approximate solution of (P_m). Let $v^m=U^m-u^m$. Then v^m satisfies

$$\begin{cases} A_mv^m\geqslant F^m-A_mu^m=R^m, \\ (u^m+v^m)_c\geqslant S^m, \\ (u^m+v^m-S^m)_c^T(A_mv^m-R^m)_c=0, \\ v^m=0 \quad \text{on } G_m\cap\Gamma_u. \end{cases} \tag{2.26}$$

On the coarser grid G_{m-1}, we solve problem (P_{m-1})

$$\begin{cases} A_{m-1}v^{m-1}\geqslant I_m^{m-1}R^m, \\ (v^{m-1}+I_m^{m-1}u^m)_c\geqslant S^{m-1}, \\ (v^{m-1}+I_m^{m-1}u^m-S^{m-1})_c^T(A_{m-1}v^{m-1}-I_m^{m-1}R^m)_c=0, \\ v^{m-1}=0 \quad \text{on } G_m\cap\Gamma_u. \end{cases} \tag{2.27}$$

Let

$$u^{m-1}=v^{m-1}+I_m^{m-1}u^m. \tag{2.28}$$

Then u^{m-1} satisfies

$$\begin{cases} A_{m-1}u^{m-1}\geqslant F^{m-1}, \\ (u^{m-1})_c\geqslant\bar{S}^{m-1}, \\ (u^{m-1}-\bar{S}^{m-1})_c^T(A_{m-1}u^{m-1}-F^{m-1})_c=0, \\ u^{m-1}=\bar{u} \quad \text{on } G_m\cap\Gamma_c, \end{cases} \tag{2.29}$$

where

$$F^{m-1}=I_m^{m-1}(F^m-A_mu^m)+A_{m-1}(I_m^{m-1}u^m),$$
$$A_{m-1}=(I_{m-1}^m)^TA_mI_{m-1}^m, \tag{2.30}$$
$$S^{m-1}=I_m^{m-1}S^m.$$

After solving u^{m-1}, we obtain a coarse-grid correction

$$u^m:=u^m+I_{m-1}^m(u^{m-1}-I_m^{m-1}u^m). \tag{2.31}$$

As to the smoothing steps, we can use the PSOR mothod (2.23). The other part of the algorithm is the same as for general elliptic partial differential equations (cf. [1]). The calculating steps run as follows:

—— Starting from $l=m$, given $u^{l,0}$, perform the PSOR iteration ν times

$$u^{l,\nu}=\mathrm{PSOR}^{\nu,\omega}(A_l,\ F^l,\ u^{l,0}). \tag{2.32}$$

—— Compute the defect

$$F^{l-1}=I_l^{l-1}(F^l-A_lu^{l,\nu}).$$

—— Compute an approximate solution u^{l-1} by solving problem $(P_{l-1})r\geqslant 1$ times with the initial value

$$u^{l-1,0}=I_l^{l-1}u^{l,\nu} \tag{2.33}$$

(when $l=0$, solve problem (P_0) exactly).

—— Interpolate and correct until $l=m$

$$u^{l,\nu}:=u^{l,\nu}+I_{l-1}^l(u^{l-1}-I_l^{l-1}u^{l,\nu}). \tag{2.34}$$

This is one MG iterative cycle and $u^{m,\nu}$ will be the initial value of the next cycle. The flow-chart is shown in Fig. 1.

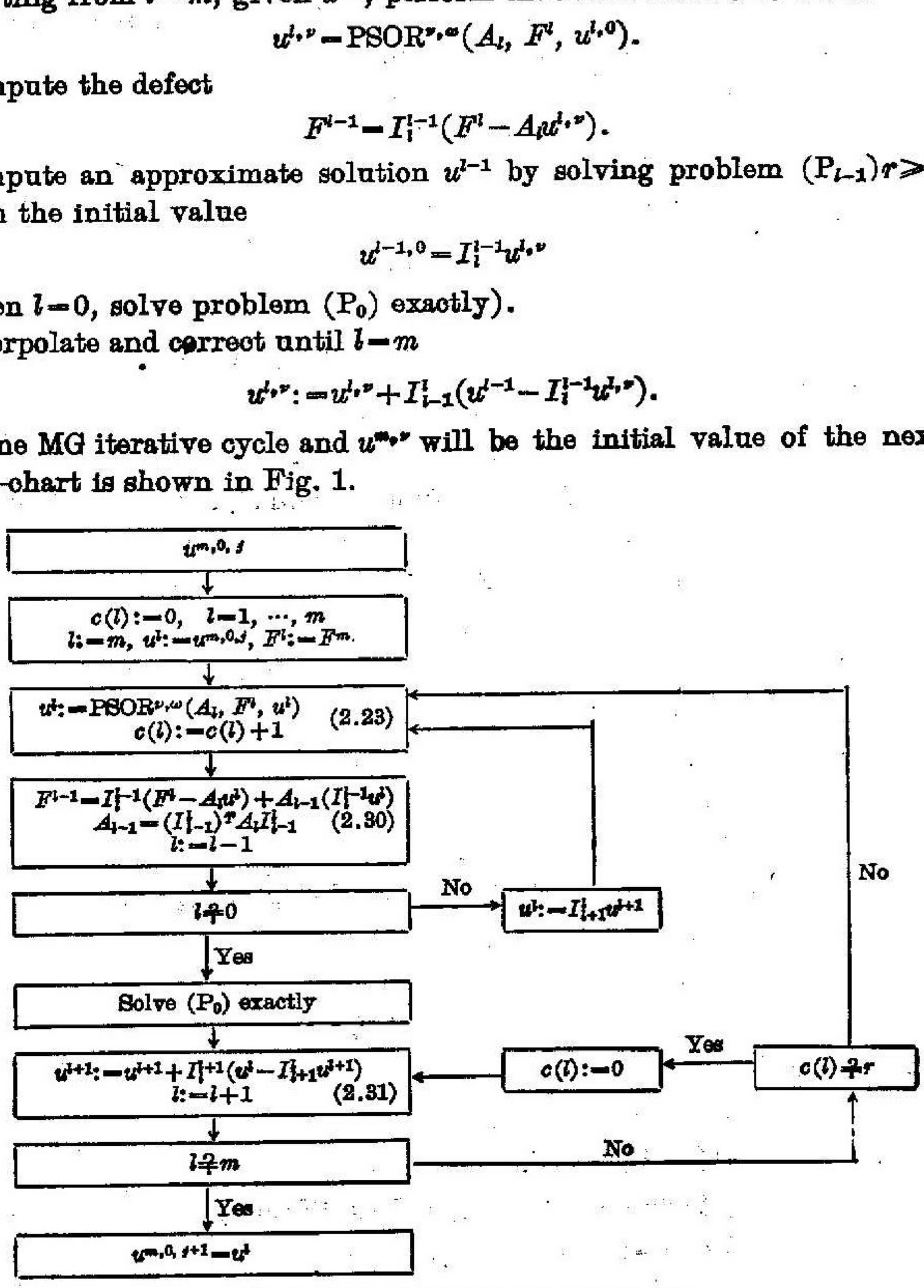

Fig. 1 Flow-chart for one multigrid cycle

Remark 1. In order to ensure $(u^m)_0 \geqslant S^m$ in (2.31), we can use a restrictive condition $(u^{m-1})_0 \geqslant S^{m-1}$ instead of $(u^{m-1})_0 \geqslant \bar{S}^{m-1}$ in (2.29), where

$$S^{m-1}=\{s_i^{m-1}\},\ s_i^{m-1}=(I_m^{m-1}u^m)_i+\max_{\alpha_{ji}>0}\{s_j^m-u_j^m,\ s_i^m-u_i^m\},\quad i,j\in I_0, \tag{2.29a}$$

$$(I_{m-1}^m u)_j=\sum_i \alpha_{ji}u_i,\quad \alpha_{ji}>0,\ \sum_i \alpha_{ji}=1$$

or use a more restrictive condition $(u^{m-1})_0 \geqslant S^{m-1}$ where

$$s_i^{m-1}=(I_m^{m-1}u^m)_i. \tag{2.29b}$$

It is easy to verify that (2.29a), (2.29b) can both ensure

$$(u^m)_0:=(u^m+I_{m-1}^m(u^{m-1}-I_m^{m-1}u^m))_0 \geqslant S^m.$$

But, calculation shows that the convergence is the best when we use the restrictive condition $(u^{m-1})_0 \geqslant \bar{S}^{m-1}$.

Remark 2. [3] proposed a slightly different algorithm, in which two rules are given to decide whether we should go to a coarser grid or return to a finer from the coarser grid. The practice of calculations shows the effects of these two algorithms are almost the same.

§ 3. The Convergency of the Algorithm

We will prove the convergency of the algorithm in this section. We use $u^{m,0,j}$ to represent the initial value of the $(j+1)$-th cycle on grid G_m. When $j=0$ let $u^{m,0,0}=0$, starting with $u^{m,0,j}$. After doing one cycle showed in Fig. 1, we evaluate $u^{m,0,j+1}=\Phi_m(u^{m,0,j})$. From the algorithm in § 2, we know $\Phi_m(u)$ is a continuous function of u. Thus (2.31), (2.33) can be respectively written as

$$\begin{aligned} u^{l,0,j+1}&=u^{l,\nu,j}+I_{l-1}^l(u^{l-1,0,j+1}-I_l^{l-1}u^{l,\nu,j})\\ u^{l-1,0,j}&=I_l^{l-1}u^{l,\nu,j},\quad l=1,2,\cdots,m,\quad j=0,1,2,\cdots. \end{aligned} \tag{3.1}$$

From the definition of J_l, it is easy to verify that

$$\begin{aligned} J_m(u^{m,0,j+1})&=J_m(u^{m,0,j})+\tilde{J}_{m-1}(u^{m-1,0,j+1}-I_m^{m-1}u^{m,\nu,j})\\ &=J_m(u^{m,\nu,j})+\tilde{J}_{m-1}((u^{m-1,\nu,j}-I_m^{m-1}u^{m,\nu,j})+I_{m-2}^{m-1}(u^{m-2,0,j+1}-I_{m-1}^{m-2}u^{m-1,\nu,j}))\\ &=J_m(u^{m,\nu,j})+\tilde{J}_{m-1}(u^{m-1,\nu,j}-I_m^{m-1}u^{m,\nu,j})+\tilde{J}_{m-2}(u^{m-2,0,j+1}-I_{m-1}^{m-2}u^{m-1,\nu,j})\\ &=\cdots=J_m(u^{m,\nu,j})+\sum_{l=1}^{m}\tilde{J}_{l-1}(u^{l-1,\nu,j}-I_l^{l-1}u^{l,\nu,j}), \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} J_l(u)&=\frac{1}{2}u^TA_lu-u^T\tilde{F}_l,\\ \tilde{F}_l&=(I_l^{l+1})^T(F^{l+1}-A_{l+1}u^{l+1}). \end{aligned} \tag{3.3}$$

According to the PSOR iteration, $J_m(u^{m,\nu,j})$ is a monotone decreasing sequence of ν[4,7], i.e.

$$J_m(u^{m,\nu,j})\leqslant J_m(u^{m,\nu-1,j})\leqslant\cdots\leqslant J_m(u^{m,0,j}). \tag{3.4}$$

It is easy to verify two identical equations below directly.

Let

$$J(u)=\frac{1}{2}u^TAu-u^TF.$$

Since A is a symmetric matrix, so for any vectors u, v, c we have

$$J(u)-J(v)=\frac{1}{2}(u-v)^TA(u-v)+(u-v)^T(Av-F),$$
$$J(u-c)-J(v-c)=\frac{1}{2}(u-v)^TA(u-v)+(u-v)^T(Av-(F+Ac)). \tag{3.5}$$

From (3.5) and PSOR iteration (2.23), we can verify easily that $\mathcal{J}_{l-1}(u^{l-1,\nu,j}-I_l^{l-1}u^{l,\nu,j})$ is a monotone decreasing function of ν in the first term $u^{l-1,\nu,j}$, i.e.

$$\mathcal{J}_l(u^{l-1,\nu,j}-I_l^{l-1}u^{l,\nu,j})\leqslant\mathcal{J}_{l-1}(u^{l-1,\nu-1,j}-I_l^{l-1}u^{l,\nu,j})\leqslant\cdots$$
$$\leqslant\mathcal{J}_{l-1}(u^{l-1,0,j}-I_l^{l-1}u^{l,\nu,j}). \tag{3.6}$$

Again from (3.1), we have

$$\mathcal{J}_{l-1}(u^{l-1,\nu,j}-I_l^{l-1}u^{l,\nu,j})\leqslant\mathcal{J}_{l-1}(0)=0,\quad l=1,\cdots,m. \tag{3.7}$$

From (3.2), (3.4) and (3.7) we have

$$J_m(u^{m,0,j+1})\leqslant J_m(u^{m,0,j}). \tag{3.8}$$

So $J_m(u^{m,0,j})$ is a monotone-decreasing function of j. Since $A_m=A$ and A is a positive definite matrix, we know that J_m has a lower bound. Thus $J_m(u^{m,0,j})$ converges to a limit J_m^*, i.e.

$$J_m(u^{m,j,j})\to J_m^*,\quad j\to\infty. \tag{3.9}$$

So $\|u^{m,0,j}\|$ is a bounded sequence and there exists a subsequence $j_1<j_2<\cdots<j_n<\cdots$ and a $\bar{u}^m$ such that

$$u^{m,0,j_n}\to\bar{u}^m,\quad n\to\infty. \tag{3.10}$$

From $u_i^{m,0,j_n}\geqslant s_i^m$, $\bar{u}_i^m\geqslant s_i^m$ is apparent $(i\in I_c)$, i.e.

$$\bar{U}_c^m=(\bar{u}_{i_1,3}^m,\cdots,\bar{u}_{i_k,3}^m)\geqslant S^m,\quad 3i_k\in I_c. \tag{3.11}$$

From (3.10) we have

$$J_m(u^{m,0,j_n})\to J_m(\bar{u}^m),\quad n\to\infty. \tag{3.12}$$

Also from the continuity of Φ_m, we have

$$u^{m,0,j_n+1}=\Phi_m(u^{m,0,j_n})\to\Phi_m(\bar{u}^m). \tag{3.13}$$

Because $J_m(u^{m,0,j})$ is the monotone decreasing function of j, we have

$$J_m(u^{m,0,j_n+1})\leqslant J_m(u^{m,0,j_n+1})\leqslant J_m(u^{m,0,j_n}).$$

So

$$J_m(u^{m,0,j_n+1})\to J_m(\bar{u}^m). \tag{3.14}$$

From (3.13), (3.14)

$$J_m(\Phi_m(\bar{u}^m))=J_m(\bar{u}^m).$$

If we denote $\bar{u}^m=\bar{u}^{m,0,0}$, then $\Phi_m(\bar{u}^m)=\bar{u}^{m,0,1}$ and the above formula is changed to

$$J_m(\bar{u}^{m,0,1})=J_m(\bar{u}^{m,0,0}). \tag{3.15}$$

Now, we prove that

$$\bar{u}^{m,1,0}=\bar{u}^{m,0,0}. \tag{3.16}$$

If not, from PSOR (2.23), we can prove easily

$$J_m(\bar{u}^{m,0,0}) > J_m(\bar{u}^{m,1,0}).$$

Additionally, from (3.2), we have

$$J_m(\bar{u}^{m,0,0}) > J_m(\bar{u}^{m,1,0}) \geqslant J_m(\bar{u}^{m,\nu,0}) \geqslant J_m(\bar{u}^{m,0,1}).$$

This contradicts (3.15). So (3.16) implies that if we take $\bar{u}^m$ as an initial value, after one PSOR iteration (2.23), it is still $\bar{u}^m$. That means $\bar{u}^m$ is one solution of (P_m). By the uniqueness of the solution, $\bar{u}^m$ is also the limit of the whole sequence $\{u^{m,0,j}\}$, i.e.

$$u^{m,0,j} \to \bar{u}^m, \quad j \to \infty. \tag{3.17}$$

Thus the convergence of the algorithm has been proved.

§ 4. Numerical Examples

We have made two general MG calculational programs MG-1, MG-2. The former is for solving rectangular domains, and adopts bi-linear elements. The latter is for general polygonal domains (the curve bound can be approached with broken-line) and adopts triangulation and linear elements. Both can solve elliptic partial differential equations and elastic mechanical equations (including linear problems and minimization problems with restrictions). We calculate the following examples:

Example 1. Solve the two dimensional linear problem (I) on a square domain $\Omega = [0, 1] \times [0, 1]$ with $h_0 = 1/2$, $N_0 = 3 \times 3 = 9$, elastic modulus $E = 7200$, and Poisson ratio $\nu = 0.3$. Suppose that the problem has an exact solution (u_1, u_2)

$$\begin{aligned} u_1(x_1, x_2) &= 100x_1(1-x_1^2)x_2(1-x_2), \\ u_2(x_1, x_2) &= 100x_1(1-x_1)x_2(1-x_2^2). \end{aligned} \tag{4.1}$$

From (2.1)—(2.3), calculate f_1, f_2,

$$u_1 = u_2 = 0, \quad (x_1, x_2) \in \partial\Omega. \tag{4.2}$$

The computing results are shown in Table 1 (by program MG-1) and Table 2 (by program MG-2), where the column "single grid" is obtained with PSOR (2.23) on the finest grid G_m. The error of the iteration is $\|s_k\| < \varepsilon$ where

$$\|s_k\| = \max_i |u_i^{(k)} - u_i^{(k-1)}|. \tag{4.3}$$

Table 1

Number of levels m	Number of nodes N_m	Error ε	Multigrid		Single grid	
			ω	WU	ω	WU
3	289	10^{-6}	1.0	56.0	1.70	112
4	1089	10^{-6}	1.0	69.1	1.80	152
5	4225	10^{-6}	1.0	61.1	1.85	390
6	16641	10^{-6}	1.0	56.2	1.90	996

Table 2

Number of levels m	Number of nodes N_m	Error e	Multigrid		Single grid	
			ω	WU	ω	WU
3	289	10^{-8}	1.0	103.6	1.70	376
4	1089	10^{-8}	1.0	154.5	1.80	1610

Remark 3.

1) In Table 1, the CPU time for $m=6$ is

$t=399''$ (multigrid), $t=6480''$ (single grid).

In Table 2, the CPU time for $m=4$ is

$t=1'30''$ (multigrid), $t=16'46''$ (single grid).

2) The unit of WU in the tables is the operating amount of running one iteration (2.23) on the finest grid G_m. So WU will be $4^{(l-m)}$ for running one iteration on G_l $(l<m)$.

3) For the MG in Table 1, WU $(m=5, 6)$ is less than WU $(m=4)$. This is because of the difference in initial error $\|e_0\|$.

Example 2. Consider the following elastic contact problem: a solid axis Ω embedded in a circular ring. Before loaded, they contact at point A. Load a horizontal concentrated force p. Simply, suppose that the initial gap $s=0$. Friction is not considered. The elements (devided for grid G_0) are shown in Fig. 2. The calculating results by program MG–2, are shown in Table 3.

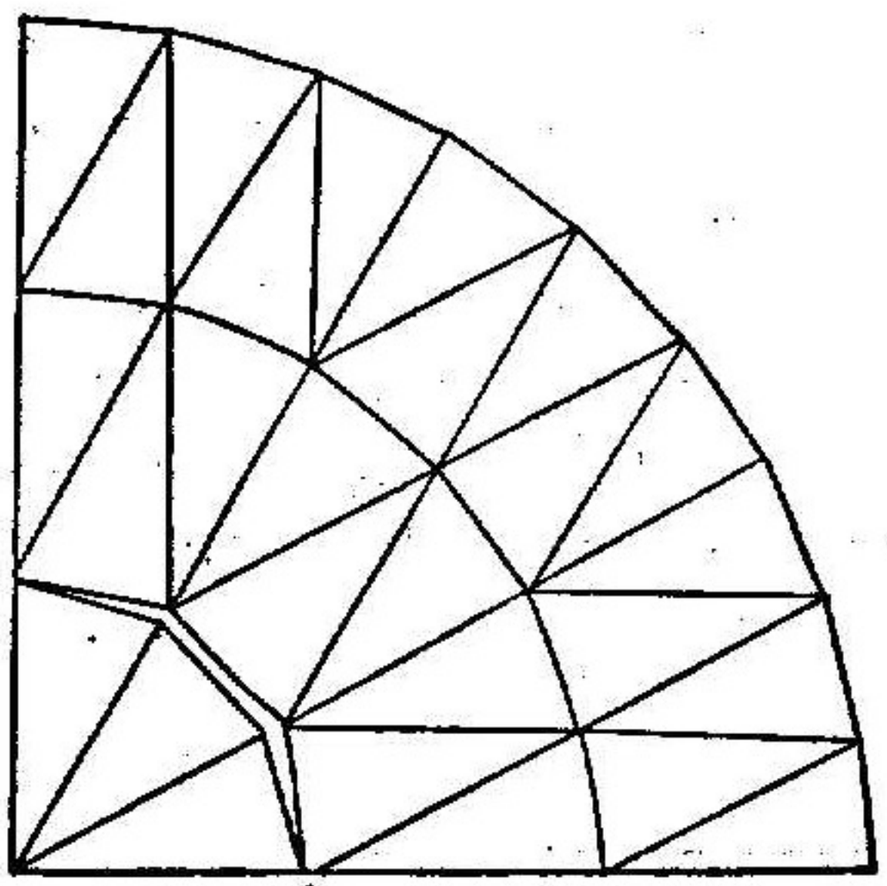

Fig. 2

Table 3

m	N_m	e	Multigrid			Single grid		
			ω	WU	CPU times	ω	WU	CPU times
2	934	10^{-6}	0.3	607.7	7'40''	195	1455	13'22''

The calculation above shows that for elastic problems, whether linear or not, the MG iteration is much more efficient than the SG iteration, especially for bi-linear elements on a rectangular domain. For elastic contact problems such as example 2, because of the inferior feature of the stiffness matrix, the iteration using SG converges very slowly, but using MG it will do much faster. Obviously the operating amount is still more than that of the linear problem which has the same size. Besides, how to choose the best ω and whether the convergence rate is independent of h for nonlinear problems are still open problems.

References

[1] K. Stüben, U. Trottenberg, Multigrid method: fundamental algorithm, model problem analysis and application, *Lecture Notes in Mathematics*, **960**, 1982.

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

[3] A. Brandt, C. W. Cryer, Multigrid algorithms for the solution of linear complementary problems arising from free boundary problems, *SIAM J. Sci. Stat. Comput.*, **4** (1983), 655—684.

[4] C. W. Cryer, The solution of a quadratic programming problem using systematic over-relaxation, *SIAM J. Control*, **9** (1971), 385—392.

[5] W. Hackbusch, On multigrid methods for variational inequalities, *Numer. Math.*, **42** (1983), 65—76.

[6] G. Duvaut, J. L. Lions, Les Inéquations en Mécanique et en Physique, Paris, 1972.

[7] R. Glowinski, J. L. Lions, R. Tremolieres, Numerical Analysis of Variational Inequalities, North-Holland, Amsterdam, 1981.

[8] Wang Jin-xian, Huang Yu-xia, Finite element approximation for the free boundary problem of the axisymmetrical stationary water cone, Proceedings of the China-France Symposium on Finite Element Method (ed. Feng Kang and J. L. Lions), Beijing, 1983.

[9] Wang Jin-xian, F. Brakhagen, Multigrid method for a type of variational inequalities, Proceedings of the 1984 Beijing symposium on Differential Geometry and Differential Equations, Beijing, 1985.

[10] G. Hadlay, Nonlinear and Dynamic Programming, Addison-Wesley, Reading, Mass., 1964.