Journal of Computational Mathematics, Vol.9, No.1, 1991, 74–85.

TWO SYNCHRONOUS PARALLEL ALGORITHMS FOR PARTIAL DIFFERENTIAL EQUATIONS*

Lu Tao
(*Chengdu Branch, Academia Sinica, Chengdu, China*)

Shih Tsi-min Liem Chin-bo
(*Hong Kong Polytechnic, Hong Kong*)

In many problems involving practical applications, e.g., in exploitation of oil fields and gas fields, nuclear engineering, large-scale structural engineering and forecasting of climate and tides, it is frequently necessary to solve initial boundary value problems of high dimensional linear or nonlinear partial differential equations defined in large regions. Solving these problems requires a large amount of computation and sometimes, even rapid responses. The idea of parallel computer system encourages numerical analysts to develop more efficient methods. It seems that the study of parallel algorithms will soon become a new target in the development of numerical analysis[2].

Based on the Schwarz algorithm, L.S. Kang[1] introduced an asynchronous parallel algorithm for solving partial differential equations of the elliptic type. In this paper, we introduce two synchronous parallel algorithms for the solution of partial differential equations. First, the region where the equations are defined is divided into some subregions which may be overlapping, and then the equations are solved in each subregion independently. Finally, a simple arithmetic mean is assigned to each point within the overlapping parts of the subregions.

The first algorithm introduced in this paper is based on the groupwise projection iterative method. We need only to solve the least-squares problem in each subregion. It can be proved that the parallel solution converges to the solution of the discrete equations for the entire region. Moreover, this result is independent of the type of differential equations. The proof of the second algorithm is based on the discrete maximum principle.

In §1 and §2 We give the descriptions of the two parallel algorithms respectively. §3 introduces the groupwise projection iterative method, and the proofs of the convergence of the two algorithms are given in §4 and §5. In §6 a numerical experiment is presented.

§1. The Parallel Algorithm 1

The boundary value problem of a partial differential equation, in general, can be written as

*Received February 6, 1988.

$$\begin{cases} Lu = f, & \text{in } \Omega, \\ lu = g, & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where Ω is a bounded region with boundary $\partial\Omega$, L is a differential operator, and l is a boundary operator. We may apply either the finite difference or the finite element method to (1) and obtain an algebraic system. For instance, consider the Dirichlet problem of a second order elliptic equation with homogeneous boundary value. We may start from its weak form

$$a(u, v) = (f, v) \quad \text{for } v \in E'(\Omega) \tag{2}$$

and then discretize by the finite element method.

Take a regular triangulation of the region Ω to get Ω_h. At each $P_j \in \Omega_h$, we define a linear shape function φ_j. Let

$$S^h = \text{ Span } \{\varphi_j, \text{ for } P_j \in \Omega_h\} \tag{3}$$

where

$$\varphi_j(P_i) = \delta_{ji}, \text{ for } \varphi_j \in S_h.$$

Hence the finite element approximation $\{u^h\}$ satisfies

$$a(u^h, \varphi_j) = (f, \varphi_j), \quad j \in I \tag{4}$$

where I is the set of indices of all grid points in Ω_h. $N(P_j) = \text{supp } (\varphi_j)$ is called the discrete neighbourhood of the point P_j. $N(P_j)$ contains only P_j and a few neighbouring points. We denote $N(P_j)$ by N_j. Obviously, (4) can be written as

$$a(u^h, \varphi_j) = \sum_{P_i \in N_j} C_{ji} u^h(P_i) = f_j, \quad \text{for } j \in I. \tag{5}$$

In other cases, such as shape functions of a higher degree, or when a finite difference method or a collocation method is used, the corresponding discrete system can be obtained in a similar way. At each grid point $P_j \in \Omega_h$, establish an equation:

$$L^h u^h(P_j) = \sum_{i \in I} C_{ji} u^h(P_i) = f_j, \quad j \in I. \tag{6}$$

We define the discrete neighbourhood as

$$N_j \equiv N(P_j) \equiv \{P_i; C_{ji} \neq 0\}$$

and (6) now becomes

$$L^h u^h(P_j) = \sum_{P_i \in N_j} C_{ji} u^h(P_i) = f_j, \quad j \in I. \tag{7}$$

In order to solve (7) by a parallel algorithm, we divide Ω_h into m subregions $\Omega_h^1, \cdots, \Omega_h^m$, $\Omega_h = \sum_{i=1}^{m} \Omega_h^i$. Some of the subregions can be overlapping. To reduce the waiting time among

different computers used in one parallel system, $\Omega_h^i, i = 1, \cdots, m$, should contain nearly the same number of grid points.

Define the discrete neighbourhood of Ω_h^i as follows:

$$N(\Omega_h^i) = \cup N(P), \quad P \in \Omega_h^i.$$

For $j \in I, P_j$ is called a k-multiple point, denoted by $P_j \in \pi_k$, if there exist at most k subregions $\Omega_h^{i_1}, \cdots, \Omega_h^{i_k}$ such that $P_j \in \bigcap_{s=1}^{k} N(\Omega_h^{i_s})$.

The procedure of the parallel algorithm 1 is as follows:

1° Choose a relaxation factor $\omega \in (0, 2)$, a tolerance $\varepsilon > 0$ and an initial $u_0 = \{u_0(P_j), j \in I\}$. Set $0 \Rightarrow n$.

2° Compute parallelly for each $\Omega_h^i, i = 1, \cdots, m$, the coefficient matrix C_{j_s} of the discrete system (for the nonlinear case, a linearization process is needed) and the residuals

$$\bar{f}_j^i = f_j - \sum_{P_s \in N_j} C_{j_s} u_n^i(P_s), \quad j \in I_i = \text{ index set of points in } \Omega_h^i$$

$$F^i = \max_{j \in I_i} |\bar{f}_j^i|, \quad i = 1, \cdots, m.$$

3° If $F = \max_{1 \le i \le m} F_i \le \varepsilon$, stop the process and output u_n; otherwise proceed to the next step.

4° Set equations for the correction Δu_n^i is each $\Omega_h^i, i = l, \cdots, m$,

$$(A_i) \qquad \sum_{P_s \in N_j} C_{j_s} \Delta u_n^i(P_s) = \bar{f}_j^i, \quad j \in I_i.$$

5° Find the least-squares solutions of $(A_i), i = l, \cdots, m$, parallelly in a parallel computer system, i.e.,

$$\Delta u_n^i = C_i^+ \bar{f}^i$$

where C_i^+ is the Moore-Penrose generalized inverse matrix of the coefficient matrix of (A_i).

6° If $P_j \in \pi_k$, then there exist $i_1, \cdots, i_k$ such that $P_j \in \bigcap_{s=1}^{k} N(\Omega_h^{i_s})$ and define

$$\Delta u_n(P_j) = \frac{1}{k} \sum_{s=1}^{k} \Delta u^{i_s}(P_j), \quad \text{for } j \in I. \tag{8}$$

7° Set $u_n + \omega \Delta u_n \Rightarrow u_n$, $n + 1 \Rightarrow n$ and go to 2°.

Remark. 1. The tolerance ε is recommended to be of order $O(h^4)$. From the posterior error estimate theorem[6], we have

$$\|u^h - u_n\|_\infty < CFK(A)$$

where C is a constant, $K(A)$ is the condition number of the discrete system in Ω_h, and F is the maximum of residuals. In general, u^h is of order $O(h^2)$, and $K(A) = O(h^{-2})$. In order to guarantee that u_n has the same order of accuracy as u^h, F should be of order $O(h^4)$.

2. In linear cases, the coefficient matrices of the systems in each subregion remain unchanged over the entire iterative process. In nonlinear cases, at each iterative step, linearization should be done first and the resulting coefficient matrices should be modified from step to step.

3. The relaxation factor $\omega \in (0, 2)$ is satisfactory for convergence. We may choose better values of ω to accelerate the convergence.

§2.The Parallel Algorithm 2

For convenience, we consider the following Dirichlet problem

$$\begin{cases} Lu = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \tag{9}$$

The discrete system in the entire region is

$$\begin{cases} L^h u_h = f^h, & \text{in } \Omega_h, \\ u_h = g, & \text{on } \partial\Omega_h. \end{cases} \tag{10}$$

Divide Ω_h into m subregions: $\Omega_h = \bigcup_{i=1}^{m} \Omega_h^i$; Ω_h^i can be overlapping. $\partial\Omega_h^i = N(\Omega_h^i) \setminus \Omega_h^i$ is the discrete boundary of Ω_h^i.

Let u^* be the unique solution of (10). (10) is equivalent to the system

$$\begin{cases} L^h u_h^i = f^h, & \text{in } \Omega_h^i, \\ u_h^i = u^*, & \text{on } \partial\Omega_h^i, \quad i = 1, \cdots, m. \end{cases} \tag{11}$$

Since u^* is unknown, (11) can only be solved by iterative methods. The procedure is as follows:

1° Choose a tolerance $\varepsilon > 0$ and an initial approximation $u_0 = \{u_0(P_j); j \in I\}$. Set $0 \Rightarrow n$.

2° Compute the residuals

$$\bar{f}_j = f^h(P_j) - L^h u_n(P_j), \quad j \in I$$

and

$$F = \max_{j \in I} |\bar{f}_j|.$$

3° If $F \le \varepsilon$, stop the process and output u_n; otherwise proceed to the next step.

4° Solve

$$\begin{cases} L^h u_{n+1}^i = f^h, & \text{in } \Omega_h^i, \\ u_{n+1}^i = u_n, & \text{on } \partial\Omega_h^i \end{cases} \tag{12}$$

for $i = 1, \cdots, m$ parallelly.

5° If $P_j \in \pi_k$, then there exist $i_1, \cdots, i_k$ such that

$$P_j \in \bigcap_{s=1}^{k} N(\Omega_h^{i_s}).$$

Set

$$u_{n+1}(P_j) = \frac{1}{k}\sum_{s=1}^{k} u_{n+1}^{i_s}(P_j); \quad j \in I.$$

6° Set $n+1 \Rightarrow n$ and go to 2°.

In §5 we will prove that u_{n+1} converges to u^* provided (12) satisfies the discrete maximum principle and $\partial\Omega_h^i \cap \partial\Omega_h \neq 0, i = 1, \cdots, m$.

§3. The Groupwise Projection Iterative Method for Linear Systems

The parallel algorithm 1 introduced in §1 is based on the groupwise projection iterative method for linear systems. The method was first established by S. Kaezmarz[3]; its further development can be found in [4]–[6]. In those papers the discussion was confined to the case of one equation in one group. To deal with our problem; we extend the method to the case of many equations contained in one group.

Consider the linear system

$$a_{j1}x_1 + \cdots + a_{jl}x_l = b_j, \quad j = 1, \cdots l. \tag{13}$$

Rewrite (13) in the form of inner product,

$$(a_j, x) = b_j, \quad j = 1, \cdots, l \tag{14}$$

and assume it has a unique solution x^*.

First the set of indices $I = \{1, 2, \cdots, l\}$ is divided into m subsets: $I = \bigcup_{i=1}^{m} I_i$, where different subsets can be overlapping. Then the system (14) is divided into m groups:

$$(B_i) \qquad (a_j, x) = b_j, \quad j \in I_i \tag{15}$$

where $i = 1, \cdots, m$.

For each i, (B_i) has at least a solution x^* which, in general, is not unique. It is well known that the least squares solution of (B_i) exists and is unique. This solution is the projection of x^* in the subspace $H_i =$ span $\{a_j; j \in I_i\}$. If we denote the projection onto H_i by E_i, then $E_i x^*$ is the least squares solution of (B_i).

First, we describe the groupwise projection iterative method.

Choose a relaxation factor $\omega (0 < \omega < 2)$ and an initial approximation $x^0 = \{x^0(P_j), j \in I\}$; then the process of getting x^{k+1} from x^k can proceed as follows:

$$x_{(s+1)}^k = x_{(s)}^k + \omega x_{(s)}^k, \quad s = 1, \cdots, 2m,$$

$$x_{(s)}^k = \{x_{(s)}^k(P_j), j \in I\} \text{ and } x_{(1)}^k = x^k,$$

where the correction $\Delta x^k_{(s)}$ is obtained from the group (B_i), i.e., $\Delta x^k_{(s)}$ is the least squares solution of the system

$$(a_j, \Delta x^k_{(s)}) = b_j - (a_j, x^k_{(s)}), \quad j \in I_i$$

where i is chosen to be the minimum of s and $2m + 1 - s$. Obviously, $\Delta x^k_{(s)}(P_j)$ is defined only for $j \in I_i$. We extend it to all $j \in I$ by simply setting $\Delta x^k_{(s)}(P_j) = 0$ for $j \in I \setminus I_i$.

Finally, we set

$$x^{k+1} = x^k_{(2m+1)}.$$

Now we are going to prove the convergence. In fact, the exact correction value of $x^k_{(s)}$ is $x^* - x^k_{(s)}$ and the least squares solution $\Delta x^k_{(s)}$ is the projection $x^* - x^k_{(s)}$ on the subspace H_i, i.e., $\Delta x^k_{(s)} = E_i(x^* - x^k_{(s)})$. Hence, from $x^k_{(s+1)} = x^k_{(s)} + \omega \Delta x^k_{(s)}$, we have

$$x^* - x^k_{(s+1)} = (I - \omega E_i)(x^* - x^k_{(s)}).$$

Let $Q_i = I - \omega E_i$. Then

$$x^* - x^{k+1} = Q_1 \cdots Q_m Q_m \cdots Q_1 (x^* - x^k) = (QQ^*)^{k+1}(x^* - x^0) \tag{16}$$

where $Q = Q_1 \cdots Q_m$. It is known, by direct computation, that $\|Q_i\| \leq 1, i = 1, \cdots, m$, and hence $\|Q\| \leq 1$. The equality holds only if there exists a vector $y, \|y\| = 1$, such that $Qy = y$. This means that y is orthogonal to all $a_j, j \in I$, and hence $y = 0$. The contradiction shows $\|Q\| = r < 1$ and

$$\|x^* - x^k\| \leq \|Q\|^{2k} \|x^* - x^0\| = r^{2k} \|x^* - x^0\|.$$

It follows that $x^k \to x^*$ as $k \to \infty$.

When the groupwise projection iterative method is applied to nonlinear systems, the convergence also follows.

Consider a nonlinear system

$$f_j(x_1, \cdots, x_l) = 0, j = 1, \cdots, l, \tag{17}$$

or its vector form

$$F(x) = 0. \tag{18}$$

Divide (17) into m groups

(C_i) $$f_j(x_1, \cdots, x_n) = 0, \quad j \in I_i$$

where $i = 1, \cdots, m$. The method can be described as follows:

Choose a relaxation factor $\omega (0 < \omega < 2)$ and an initial approximation x^0. Then for $s = 0, 1, 2, \cdots, 2m$, let successively

$$x^k_{(s+1)} = x^k_{(s)} + \omega x^k_{(s)} \tag{19}$$

where the correction $\Delta x^k_{(s)}$ is the least squares solution of the system

(D_i) $$(\text{ grad } f_j(x^k), \Delta x^k_{(s)}) = -f_j(x^k), \quad j \in I_i \tag{20}$$

where $i = \min(s, 2m + 1 - s)$ and $x^k_{(0)} = x^k$. Set

$$x^{k+1} = x^k_{(2m+1)}. \tag{21}$$

We shall prove the convergence of (19)–(21). We have, from the previous derivation, that

$$x^{k+1} = x^k - (I - S_k S_k^*)[F'(x^k)]^{-1} F(x^k) \tag{22}$$

where $S_k = Q_1(x^k) \cdots Q_m(x^k)$ and $Q_i(x^k) = I - \omega E_i(x^k), i = 1, \cdots, m$ and $E_i(x^k)$ is the projection of x^k into the subspace Span $(\text{grad} f_j(x^k); j \in I_i)$. From $\|S_k S_k^*\| < 1$, we know $(I - S_k S_k^*)$ is invertible. Let $G = I - (I - SS^*) = SS^*$, where $S = Q_1(x^*) \cdots Q_n(x^*)$. Then the groupwise projection iterative method is convergent since the Newton iterative method is convergent under the condition $\rho(SS^*) < 1$.

§4. The Proof of Algorithm 1

In the following, we only prove the linear case. The nonlinear case can be proved in a similar way.

Let

$$\sum_{S \in N_j} C_{js} u_s = f_j, \quad j \in I, \quad u_s = u^h(P_s) \tag{23}$$

($s \in N_j$ means $P_s \in N_j$) be the discrete system defined in the entire region Ω_h and

$$(E_i) \qquad \sum_{s \in N_j} C_{js} u_s^i = f_j, \quad j \in I_i, \quad i = 1, \cdots, m \tag{24}$$

be the discrete system defined in the ith subregion Ω_h^i.

When we solve (E_i) by using the parallel algorithm 1, we take into account that $u_s^{i_1}$ and $u_s^{i_2} (i_1 \neq i_2)$ are independent. From this point of view, we have assumed that (E_{i_1}) and (E_{i_2}) have no unknowns in common.

As a compensation to this assumption, we add the following extra restrictions to (E_i),

$$(F_s) \qquad \begin{cases} u_s^{i_1} - u_s^{i_2} = 0, \\ \cdots\cdots \\ u_s^{i_1} - u_s^{i_k} = 0, \end{cases} \quad \text{for } P_s \in \pi_k, \quad k \geq 2. \tag{25}$$

Obviously, the equations $(E_i), i = 1, \cdots, m$, togeter with the equations (F_s), $P_s \in \pi_k, k \geq 2$, are equivalent to equations (23).

We name $(E_i), i = 1, \cdots, m$, as group 1 and $(F_s), P_s \in \pi_k, k \geq 2$, as group 2. Since (E_{i_1}) and (E_{i_2}) $(i_1 \neq i_2)$ have no unknowns in common, the least squares solution of group 1 is simply the union of the least squares solutions of (E_i), $i = 1, \cdots, m$, and the latter can be found parallelly.

Let $u_0 = u_0(P_j), j \in I$, be an initial approximation to $u^* = \{u^h(P_j), j \in I\}$, which is the solution of (23). Let $\Delta u_n^i = \{\Delta u_n^i(P_j), j \in I_i\}, i = 1, \cdots, m$, be the correction of u_n obtained from the least squares solution of group 1 and let $\delta u_n^i(P_s), P_s \in \pi_k, k \geq 2$, be the correction of $u_n^i + \Delta u_n^i$ obtained from the least squares solution of group 2. Noting that when $s_1 \neq s_2, (F_{s_1})$ and (F_{s_2}) have no unknowns in common, we may find the least squares solutions of (F_s), $P_s \in \pi_k$, $k \geq 2$, parallelly. If $P_s \in \pi_k, k \geq 2$, then there exist $\Omega_h^{i_1}, \cdots \Omega_h^{i_k}$

such that $P_s \in \bigcap_{j=1}^{k} N(\Omega_h^{ij})$. Substituting $u_n^i + \Delta u_n^i + \delta u_n^i$ into (F_s), we have

$$(G_2) \qquad \begin{cases} \delta u_n^{i_1}(P_s) - \delta u_n^{i_2}(P_s) = -(\Delta u_n^{i_1}(P_s) - \Delta u_n^{i_2}(P_s)), \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \delta u_n^{i_1}(P_s) - \delta u_n^{i_k}(P_s) = -(\Delta u_n^{i_1}(P_s) - \Delta u_n^{i_k}(P_s)). \end{cases}$$

The general solution of (F_s) is $C - \Delta u_n^{ij}(P_s), j = 1, \cdots, k$, where C is an arbitrary constant. Finding the least squares solution of (F_s) is equivalent to determining C which minimizes $\sum_{j=1}^{k}(C - \Delta u_n^{ij}(P_s))^2$. Obviously,

$$C = \frac{1}{k}\Big(\sum_{j=1}^{k} \Delta u_n^{ij}(P_s)\Big)$$

and

$$\delta u_n^{ij}(P_s) = \frac{1}{k}\Big(\sum_{j=1}^{k} \Delta u_n^{ij}(P_s)\Big) - \Delta u_n^{ij}(p_s).$$

Finally, we have

$$u_{n+1}(P_s) = u_n(P_s) + \frac{1}{k}\Big(\sum_{j=1}^{k} \Delta u_n^{ij}(P_s)\Big), \quad P_s \in \pi_k, \quad k \geq 1.$$

Let E_1, E_2 be the projections defined by group 1 and group 2 respectively. Let $Q_i = I - \omega E_i, i = 1, 2$, and $Q = Q_1 Q_2$. According to the proof given in §3, we conclude that

$$\|u^* - u_n\| = \|(QQ^*)^n(u^* - u_0)\| < r^{2n}\|u^* - u_0\|$$

where

$$r = \|Q\| = \|(I - \omega E_1)(I - \omega E_2)\| < 1.$$

When $\omega = 1, Q = E_1^{\perp} E_2^{\perp}$ and

$$r = \|E_1^{\perp} E_2^{\perp}\| = \sup_{\|u\|=1} \|E_1^{\perp} E_2^{\perp} u\| = \sup_{\substack{u \in H_2^{\perp} \\ \|u\|=1}} \|E_1^{\perp} u\| = \sup_{\substack{u \in H_2^{\perp} \\ \|u\|=1}} \sup_{\substack{v \in H_1^{\perp} \\ \|v\|=1}} |(u, v)|$$

$$= \cos(\widehat{H_1^{\perp}, H_2^{\perp}}) = \cos(\widehat{H_1, H_2})$$

where H_1 and H_2 are subspaces formed by group 1 and group 2 respectively, and $\cos(\widehat{H_1, H_2})$ is the cosine of the angle between subspaces H_1 and H_2.

§5. The Proof of Algorithm 2

We know, from (11) and (12), that $u^i_{n+1} - u^*$ satisfies

$$\begin{cases} L^h(u^i_{n+1} - u^*) = 0, & \text{in } \Omega^i_h, \\ u^i_{n+1} - u^* = u_n - u^*, & \text{on } \partial\Omega^i_h \setminus \partial\Omega_h, \\ u^i_{n+1} - u^* = 0, & \text{on } \partial\Omega^i_h \cap \partial\Omega_h, \end{cases} \tag{27}$$

where $u^i_{n+1} = \{u^i_{n+1}(P); P \in N(\Omega^i_h)\}$, $\partial\Omega^i_h \cap \partial\Omega_h \neq 0$.

Let

$$u_{n+1} = (\bar{u}^i_{n+1}(P); P \in N(\Omega^i_h), i = 1, \cdots, m\},$$

where $\bar{u}^i_{n+1}(P)$ is defined as follows: if $P \in \pi_k$, there exist $i_1, \cdots, i_k$ such that $P \in \bigcap_{j=1}^{k} N(\Omega^{i_j})$ and $\bar{u}^i_{n+1}(P) = \frac{1}{k}\sum_{j=1}^{k} u^{i_j}_{n+1}$.

We now prove

$$u_n - u^* \to 0, \text{ as } n \to \infty.$$

First, we prove that the sequence $\|u^* - u_n\|_\infty$ in monotonically decreasing, i.e., if $\|u^* - u_n\|_\infty \neq 0$, then $\|u^* - u_{n+1}\|_\infty < \|u^* - u_n\|_\infty$. In fact, if $|u^* - u_{n+1}|$ at $P \in \pi_1$ attains its maximum value, then there exists Ω^i_h such that $P \in \Omega^i_h$. Applying the maximum principle to (27), it follows that $u^* - u_{n+1}$ is constant for all $P \in N(\Omega^i_h)$. By the condition $\partial\Omega^i_h \cap \partial\Omega_h \neq 0$ and $u^* - u_{n+1} = 0$ on $\partial\Omega^i_h \cap \partial\Omega_h$, we deduce that $\|u^* - u_{n+1}\|_\infty = |u^*(P) - u_{n+1}(P)| = 0$.

If $|u^* - u_{n+1}|$ at $P \in \pi_k (k \geq 2)$ attains the maximum, then

$$u_{n+1}(P) = \frac{1}{k}\sum_{j=1}^{k} u^{i_j}_{n+1}(P)$$

and

$$|u^*(P) - u_{n+1}(P)| \leq \frac{1}{k}\sum_{j=1}^{k} |u^*(P) - u^{i_j}_{n+1}(P)| \leq \max_{1\leq j\leq k} |u^*(P) - u^{i_j}_{n+1}(P)|. \tag{28}$$

The equality $|u^*(P) - u_{n+1}(P)| = \max_{1\leq j\leq k} |u^*(P) - u^{i_j}_{n+1}(P)|$ holds only if $|u^*(P) - u^{i_j}_{n+1}(P)|$, $j = 1, \cdots, k$, are equal. In this case, there exists a subregion $\Omega^{i_j}_h$ such that $P \in \Omega^{i_j}_h$ and $\|u^* - u_{n+1}\|_\infty = 0$ follows according to the previous discussion.

Now consider the case

$$|u^*(P) - u_{n+1}(P)| < \max_{1\leq j\leq k} |u^*(P) - u^{i_j}_{n+1}(P)|. \tag{29}$$

Without loss of generality, we assume that $|u^*(P) - u^{i_j}_{n+1}(P)|, j = 1, \cdots, k$, attains a maximum when $j = 1$. Then,

$$\begin{aligned} \|u^* - u_{n+1}\|_\infty &= |u^*(P) - u_{n+1}(P)| < |u^*(P) - u^{i_1}_{n+1}(P)| \\ &\leq \max_{Q\in\Omega^{i_1}_h} |u^*(Q) - u^{i_1}_{n+1}(Q)| = \max_{Q\in\partial\Omega^{i_1}_h} |u^*(Q) - u^{i_1}_n(Q)| \leq \|u^* - u_n\|_\infty. \end{aligned} \tag{30}$$

Let S be an operator such that

$$u_{n+1} - u^* = S(u_n - u^*).$$

(30) concludes that $\{u_{n+1} - u^*\}$ has a convergent subsequence.

For convenience, we still write the subsequence as $u_{n+1} - u^*$ and its limit as $v - u^*$, i.e.,

$$u_{n+1} - u^* \to v - u^*, \qquad \text{as } n \to \infty.$$

Hence, we have

$$v - u^* = s(v - u^*) \tag{31}$$

and

$$\|v - u^*\|_\infty = \|S(v - u^*)\|_\infty < \|v - u^*\|_\infty. \tag{32}$$

The inequality in (32) is obtained by applying (30) to (31). It follows, from (32), that $v = u^*$.

We have proved that the sequence $\{\|u_n - u^*\|_\infty\}$ has a limit and that there is a subsequence of $\{\|u_n - u^*\|_\infty\}$ which has a limit zero; hence we can reduce that

$$\|u_n - u^*\|_\infty \to 0 \quad \text{as } n \to \infty,$$

and the convergence of the parallel algorithm 2 is obtained.

Remark. In using the five-point finite difference method to a 2nd-order elliptic equation, it is well known that the discretized system satisfies the maximum principle. When a finite element method is employed, then, in order to satisfy the maximum principle, the triangulation of Ω should satisfy some conditions (see [8] and [9]).

§6. A Numerical Example

1° Equation:

$$\Delta u = 0, \qquad \text{in } \Omega,$$

$$u = \frac{1}{2\pi} \arctan \frac{y}{x}, \qquad \text{in } \partial\Omega$$

where $\Omega = \{-1 < x, y < 1\}/\{0 \le x < 1, y = 0\}$ (see Fig. 1).

2° Exact solution:

$$u = \frac{1}{2\pi} \arctan \frac{y}{x}.$$

3° Scheme used:

ordinary 5-point scheme, i.e.

$$\Delta u \cong \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{h^2}.$$

4° Step size:

$$h = 0.2$$

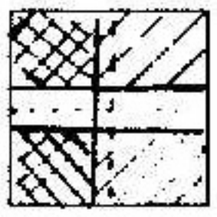
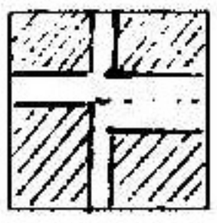
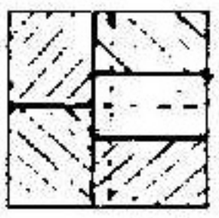
5° Method description:

First, Ω is covered by a uniform mesh of spacing $h = \frac{1}{5}$, and at each interior grid point the equation is discretized by the five-point finite difference method. At the point $(0,0)$, u is assigned three different values $0.5, 0.75$ and 0.25 where $(0,0)$ is on the right, the lower and the upper positions respectively of the five-point stencil $\Delta_h u = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix} u$. This is acceptable since the theoretical solution itself is discontinuous at the point (0,0).

Then, the resulting linear system is grouped into m systems with a smaller size, which corresponds to dividing Ω into m subregions. The system is solved by parallel algorithms 1 and 2 respectively.

The numerical results corresponding to $m = 76$, 3 and 4 are given in Table 1. The calculations were carried out on the VAX 750 computer at Hong Kong Polytechnic and the double precision arithmetic was used.

Table 1

Program	Method	Number of subregions	Size of the largest matrix
Paral 1	finite difference	1	76×76
Paral 2	parallel algorithm 1	76	1×1
Paral 3	parallel algorithm 1		36×36
Paral 4	parallel algorithm 2		36×36
Paral 5	parallel algorithm 1		20×20
Paral 6	parallel algorithm 2		20×20
Paral 7	parallel algorithm 1		24×24
Paral 8	parallel algorithm 2		24×24

Continued

Program	Number of iterations	Computer CUP (sec.)	Accuracy $\max \lvert u - u_h \rvert$
Paral 1	1	11.92	0.1781×10^{-2}
Paral 2	304	40.20	0.1953×10^{-2}
Paral 3	68	31.71	0.1793×10^{-2}
Paral 4	11	10.00	0.1782×10^{-2}
Paral 5	246	52.45	0.1893×10^{-2}
Paral 6	29	10.68	0.1786×10^{-2}
Paral 7	127	34.39	0.1844×10^{-2}
Paral 8	19	10.46	0.1783×10^{-2}

6° Conclusions: From the results in Table 1 the following general comments can be made:

a) It is found that with a given fixed error tolerance $\epsilon = 5 \times 10^{-6}$, almost exactly the same degree of accuracy was obtained for all methods.

b) By comparing the number of iterations, it is shown that the parallel algorithm 2 gives much better convergence rates than the parallel algorithm 1.

c) It is observed that when the overlappling parts of subregions enlarge, the number of iterations needed to obtain the numerical solution to the desired degree of accuracy will decrease. In this case, certainly, the size of coefficient matrices involved become larger and hence the computer time needed for each iteration will increase. Therefore, it is difficult to draw a general criterion for how to get the optimal subdivision of Ω with regard to both computing time and storage.

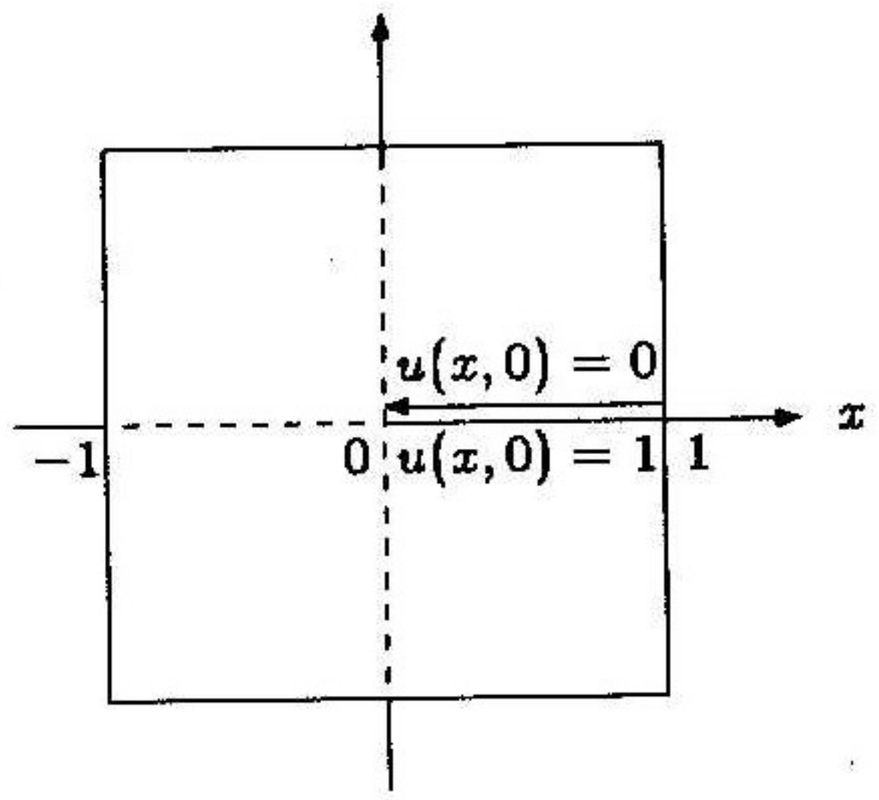


Fig. 1

Remark. We have proved, without using any preconditional technique, that the convergence rate of the parallel algorithm is $0(1/\ln\frac{1}{h})$. This will be introduced in our next paper.

References

[1] L.S. Kang, The Schwarz Algorithm, Wuhan University, National Science Edition, Special Issue of Mathematics, China, 1981, 77–88.

[2] Richard E. Ewing, Applied mathematics in the petroleum industry, *SIAM News,* **16,** 6, 1983.

[3] S. Kaczmarz, Angenäherte Auflösung von Systemen Linear Gleichungen, *Bull. Internat. Acad. Polon. Science et Lettres,* 1937, 355–357.

[4] A. Björck and T. Elfving, Accelerated Projection Method for Computation of Pseudoinverse Solutions of Systems of Linear Equations, *BIT,* **19,** 1979, 143–163.

[5] Liu Jia-quan, Convergence of SOR projection method for nonlinear systems, *J. Sys. Sci. & Math. Sci.,* **1** (1) (1981), 77–79.

[6] Lu Tao and Lin Qun, A projection iterative method for solving integral equations of the first kind and algebraic systems, *Math. Numer. Sinica,* **6 : 2** (1984), 113–120.

[7] J.M. Ortega, Numerical Analysis, Computer Science and Applied Mathematics, Academic Press, New York, 1973.

[8] G. Strang and G.J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, 1973.

[9] Tsutomu Ikeda, Maximum Principle in Finite Element Models for Convection-Diffusion Phenomena, North-Holland, 1983.

[10] J.H. Bramble, J.E. Pasciak and A.H. Schatz, An iterative method for elliptic problems on regions partitioned into substructures, *Math. Comp.,* **46 : 173,** (1986) 361–369.