J. Univ. Kuwait (Sci.) **16**, 1989

Asynchronous algorithms for heat conduction in composite media

M. N. EL-TARAZI AND M. N. ANWAR†

Department of Mathematics, University of Kuwait, P.O. Box 5969, Safat 13060, Kuwait

ABSTRACT

The problem of two dimensional steady-state heat conduction in a medium composed of two homogeneous and isotropic, but physically dissimilar materials is discretized with the method of lines to obtain a system of second order differential equations with multipoint boundary conditions. This differential system is converted, using invariant imbedding for each one dimensional problem, into a fixed point problem and then sequential and parallel asynchronous algorithms are applied.

1. INTRODUCTION

Many engineering and physical applications depend on material composed of two or more constituents. Much research has been done and many solutions have been obtained for elasticity and plasticity problems related to such materials. The subject of heat conduction in composite media, however, has received less attention.

In this paper we treat the problem of two dimensional steady-state heat conduction in a medium composed of two homogeneous and isotropic, but physically dissimilar materials. In situations where the boundary conditions are not simple elementary functions the exact analytical solution of such problems is impossible even for regions having a simple geometry. Thus a numerical technique is needed and the authors propose in this paper, (as they have done in Anwar & El-Tarazi (1985) for Poisson's equation with nonlinear boundary conditions), to discretize the considered heat conduction problem by the method of lines (Meyer 1978) to obtain a system of second order differential equations with multipoint boundary conditions. This differential system is converted, using invariant imbedding (Meyer 1973) for each one dimensional problem, into a fixed point problem for which several iterative algorithms can be used. If the asynchronous parallel algorithms are used, we obtain an approach which combines both simplicity (referring to the classical invariant imbedding associated with the by lines approximation) and efficiency (referring to the asynchronous parallel algorithms).

In Section 2, we give the mathematical formulation of the problem considered in this paper together with the appropriate boundary and auxiliary conditions. In

† Present address: Department of Engineering Mathematics, Faculty of Engineering, University of Alexandria, El-Hadhra, Alexandria, Egypt.

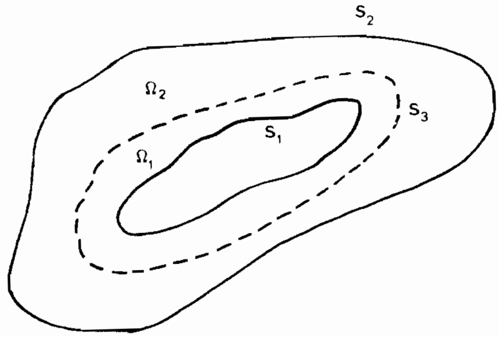

Fig. 1. Schematic representation of the domains of the problem

Section 3, we describe briefly the method of lines and invariant imbedding, and its adaptation to the present problem. In Section 4, we describe the associated fixed point problem and several sequential and parallel iterative algorithms. In the last section, we apply several sequential algorithms and simulate parallel computation for a numerical example. The results obtained show that parallel algorithms can considerably accelerate classical sequential-type algorithms.

2. MATHEMATICAL FORMULATION

Consider two continua of different thermal conductivities κ_1 and κ_2 with an interface S_3. Let the two domains occupied by the materials be denoted by Ω_1 and Ω_2 with boundaries S_1 and S_2 respectively (Fig. 1).

For the steady heat conduction problem, the temperature distributions T_1 and T_2 in the materials occupying the domains Ω_1 and Ω_2 respectively, satisfy the Laplace equations (Jaworski 1981), given by

$$\Delta T_1 = 0 \quad \text{in} \quad \Omega_1 \tag{2.1.1}$$

$$\Delta T_2 = 0 \quad \text{in} \quad \Omega_2 \tag{2.1.2}$$

The problem is supplemented by the appropriate boundary conditions along the boundary S_1 and S_2. For the sake of simplicity, we assume the temperature to be known along the boundary, that is

$$T_1 = A_1 \quad \text{along} \quad S_1 \tag{2.2.1}$$

$$T_2 = A_2 \quad \text{along} \quad S_2 \tag{2.2.2}$$

where A_1 and A_2 are given functions on S_1 and S_2 respectively. Additional conditions are needed along the interface S_3 separating the two domains. These represent the coupling conditions, namely

$$T_1 = T_2 \qquad \text{along} \quad S_3 \tag{2.3.1}$$

$$\kappa_1 \frac{\partial T_1}{\partial n_1} = \kappa_2 \frac{\partial T_2}{\partial n_1} \quad \text{along} \quad S_3 \tag{2.3.2}$$

where n_1 is the outward normal to the boundary of Ω_1.

It should be mentioned that the boundary conditions (2.2) are of Dirichlet type and other types of boundary conditions such as Newmann or mixed can also be considered.

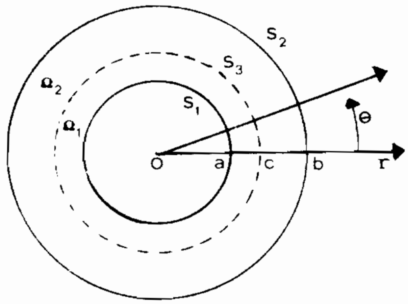

Fig. 2. Cylindrical annulus form of the composite media.

3. THE METHOD OF LINES AND INVARIANT IMBEDDING

To simplify the exposition of the numerical technique to be used for the solution of this problem, we consider a simpler geometry for the two domains Ω_1 and Ω_2; we assume that the composite medium is of a cylindrical annulus form. Furthermore we assume no variation in temperature along its axis. Hence, taking a cross section normal to its axis, we obtain three concentric circles S_1, S_2 and S_3 with radii a, b and c respectively. Then Ω_1 is the region bounded by the two circles S_1 and S_3, and Ω_2 will be the region bounded by the two circles S_3 and S_2 as shown in Fig. 2. Therefore, using the polar coordinates r and θ, the working equations are

$$\frac{\partial^2 T_1}{\partial r^2} + \frac{1}{r}\frac{\partial T_1}{\partial r} + \frac{1}{r^2}\frac{\partial^2 T_1}{\partial \theta^2} = 0 \quad \text{in} \quad \Omega_1 \tag{3.1.1}$$

$$\frac{\partial^2 T_2}{\partial r^2} + \frac{1}{r}\frac{\partial T_2}{\partial r} + \frac{1}{r^2}\frac{\partial^2 T_2}{\partial \theta^2} = 0 \quad \text{in} \quad \Omega_2 \tag{3.1.2}$$

The boundary conditions are

$$T_1 = A_1 \quad \text{along} \quad S_1 \tag{3.2.1}$$

$$T_2 = A_2 \quad \text{along} \quad S_2 \tag{3.2.2}$$

The interface conditions are

$$T_1 = T_2 \qquad \text{along} \quad S_3 \tag{3.3.1}$$

$$\frac{\partial T_1}{\partial r} = +\kappa \frac{\partial T_2}{\partial r} \quad \text{along} \quad S_3 \tag{3.3.2}$$

where $\kappa = \kappa_2/\kappa_1$ is a negative real number.

3.1. THE METHOD OF LINES

The method of lines for elliptic and parabolic problems has been described and analysed by Meyer (1973, 1978). We will adapt it to the above heat conduction problem. Let us introduce the equidistant rays making angles θ_k, $0 = \theta_0 < \theta_1 < \cdots < \theta_n = 2\pi$ with the horizontal axis and separated by the increment $\Delta\theta = \theta_{k+1} - \theta_k$ for $k = 0, 1, \ldots, n-1$. In the method of lines approximations, we retain one independent variable, say r, continuous, and discretize all the terms involved in the problem with respect to the other variable θ. If $T_{1,k}$ and $T_{2,k}$ denote the solution of (3.1) to (3.3) along the k-th ray, then the method of lines approximation for the problem (3.1) to (3.3) gives, for $k = 1, 2, \ldots, n$

$$T''_{1,k} + \frac{1}{r} T'_{1,k} + \frac{1}{(r\,\Delta\theta)^2}(T_{1,k-1} - 2T_{1,k} + T_{1,k+1}) = 0 \tag{3.4.1}$$

$$T''_{2,k} + \frac{1}{r} T'_{2,k} + \frac{1}{(r\,\Delta\theta)^2}(T_{2,k-1} - 2T_{2,k} + T_{2,k+1}) = 0 \tag{3.4.2}$$

$$T_{1,k}(a) = A_{1,k} \tag{3.4.3}$$

$$T_{2,k}(b) = A_{2,k} \tag{3.4.4}$$

$$T_{1,k}(c) = T_{2,k}(c) \tag{3.4.5}$$

$$\frac{\partial T_{1,k}}{\partial r}(c) = +\kappa \frac{\partial T_{2,k}}{\partial r}(c) \tag{3.4.6}$$

The periodicity of the problem implies $T_{1,0} = T_{1,n}$ and $T_{2,0} = T_{2,n}$. The above system represents a second order two point value problem that can be attacked by a variety of methods. We suggest the application of the method of invariant imbedding due to its simplicity.

3.2. THE INVARIANT IMBEDDING

This method (Bellman 1975; Meyer 1973, 1978) proceeds as follows: the unknown functions $T_{1,k}$, $T'_{1,k}$, $T_{2,k}$ and $T'_{2,k}$ along the k-th ray are related through the Riccati transformations

$$T_{1,k}(r) = R_{1,k}(r)T'_{1,k}(r) + W_{1,k}(r) \tag{3.5.1}$$

$$T_{2,k}(r) = R_{2,k}(r)T'_{2,k}(r) + W_{2,k}(r) \tag{3.5.2}$$

Substituting (3.5.1) [(3.5.2)] into (3.4.1) [(3.4.2)], the problem (3.4) becomes equivalent to the system

$$R'_{1,k}(r) = 1 + \frac{1}{r} R_{1,k}(r) - \frac{2}{(r\,\Delta\theta)^2} R^2_{1,k}(r),\ R_{1,k}(a) = 0 \tag{3.6.1}$$

$$\begin{aligned} W'_{1,k}(r) = &- \frac{2}{(r\,\Delta\theta)^2} R_{1,k}(r)W_{1,k}(r) \\ &+ \frac{1}{(r\,\Delta\theta)^2} R_{1,k}[T_{1,k-1}(r) + T_{1,k+1}(r)],\ W_{1,k}(a) = A_{1,k} \end{aligned} \tag{3.6.2}$$

$$R'_{2,k}(r) = 1 + \frac{1}{r} R_{2,k}(r) - \frac{2}{(r\,\Delta\theta)^2} R^2_{2,k}(r),\ R_{2,k}(b) = 0 \tag{3.6.3}$$

$$\begin{aligned} W'_{2,k} = &- \frac{2}{(r\,\Delta\theta)^2} R_{2,k}(r)W_{2,k}(r) \\ &+ \frac{1}{(r\,\Delta\theta)^2} R_{2,k}(r)[T_{2,k-1}(r) + T_{2,k+1}(r)],\ W_{2,k}(b) = A_{2,k} \end{aligned} \tag{3.6.4}$$

$$R_{1,k}(c)T'_{1,k}(c) + W_{1,k}(c) = R_{2,k}(c)T'_{2,k}(c) + W_{2,k}(c) \tag{3.6.5}$$

$$T'_{1,k}(c) = +\kappa T'_{2,k}(c) \tag{3.6.6}$$

$$[T'_{1,k}(r)]' + T'_{1,k}(r) + \frac{1}{(r\,\Delta\theta)^2} \times \{T_{1,k-1}(r) - 2[R_{1,k}(r)T'_{1,k}(r) + W_{1,k}(r)] + T_{1,k+1}(r)\} = 0 \quad (3.6.7)$$

$$[T'_{2,k}(r)]' + \frac{1}{r}\,T'_{2,k}(r) + \frac{1}{(r\,\Delta\theta)^2} \times \{T_{2,k-1}(r) - 2[R_{2,k}(r)T'_{2,k}(r) + W_{2,k}(r)] + T_{2,k+1}(r)\} = 0 \quad (3.6.8)$$

Eqns (3.6.1) and (3.6.2) are integrated forward from $r = a$ to $r = b$ while Eqns (3.6.3) and (3.6.4) are integrated backward from $r = b$ to $r = c$. Then we compute the values of $T'_{1,k}(c)$ and $T'_{2,k}(c)$ from Eqns (3.6.5) and (3.6.6). Hence (3.6.7) [resp. (3.6.8)] becomes an initial value problem which is integrated, when an iterative method is used, backward from $r = c$ to $r = a$ [resp. forward from $r = c$ to $r = b$]. Finally (3.5.1) and (3.5.2) are used to compute $T_{1,k}$ and $T_{2,k}$ respectively.

4. ASSOCIATED FIXED POINT PROBLEM AND ITERATIVE ALGORITHMS

The problem by lines approximations (3.6.1) to (3.6.8) can be solved by Jacobi, Gauss Seidel or over relaxation type methods. More general algorithms, namely the asynchronous algorithms which allow parallel computation for the problem, can be considered.

4.1. FIXED POINT PROBLEM ASSOCIATED WITH (3.6)

We convert the discretized problem (3.6.1) to (3.6.8) into a fixed point problem of an operator G which will be defined in what follows:

First, for $k = 1, 2, \ldots, n$ let

$$\begin{cases} E_{1,k} = \{\phi_{1,k} \in C^2[a, c],\ \phi_{1,k}(a) = A_{1,k}\} \\ E_{2,k} = \{\phi_{2,k} \in C^2[c, b],\ \phi_{2,k}(b) = A_{2,k}\} \end{cases} \quad (4.1)$$

and the product space

$$E = E_{1,1} \times \cdots \times E_{1,n} \times E_{2,1} \times \cdots \times E_{2,n} \quad (4.2)$$

We now define the operator

$$\begin{cases} G : E \to E \\ v = (v_{1,1}, \ldots, v_{1,n}, v_{2,1}, \ldots, v_{2,n}) \to u = (u_{1,1}, \ldots, u_{1,n}, u_{2,1}, \ldots, u_{2,n}) \end{cases} \quad (4.3)$$

where for any $k \in \{1, 2, \ldots, n\}$ $u_{1,k}$ and $u_{2,k}$ are the solutions in $E_{1,k}$ and $E_{2,k}$ respectively to

$$R'_{i,k}(r) = 1 + \frac{1}{r}\,R_{i,k}(r) - \frac{2}{(r\,\Delta\theta)^2}\,R^2_{i,k}(r),\quad R_{i,k}(a_i) = 0 \quad (4.4.1)$$

$$W'_{i,k}(r) = -\frac{2}{(r\,\Delta\theta)^2}\,R_{i,k}(r)W_{i,k}(r) + \frac{1}{(r\,\Delta\theta)^2}\,R_{i,k}[v_{i,k-1}(r) + v_{i,k+1}(r)],\quad W_{i,k}(a_i) = A_{i,k} \quad (4.4.2)$$

$$R_{1,k}(c)u'_{1,k}(c) + W_{1,k}(c) = R_{2,k}(c)u'_{2,k}(c) + W_{2,k}(c) \tag{4.4.3}$$

$$u'_{1,k}(c) = +\kappa u'_{2,k}(c) \tag{4.4.4}$$

$$\begin{cases} [u'_{i,k}(r)]' + \dfrac{1}{r} u'_{i,k}(r) + \dfrac{1}{(r\,\Delta\theta)^2} \\ \times \{v_{i,k-1}(r) - 2[R_{i,k}(r)u'_{i,k}(r) + W_{i,k}(r)] + v_{i,k+1}(r)\} = 0 \\ u'_{i,k}(c) \text{ is computed from (4.4.3) and (4.4.4)} \end{cases} \tag{4.4.5}$$

and then u_{ik} is computed from

$$u_{i,k}(r) = R_{i,k}(r)u'_{i,k}(r) + W_{i,k}(r) \tag{4.4.6}$$

where $i = 1, 2$, $a_1 = a$ and $a_2 = b$.

It is clear that $u^* \in E$ is a solution to the problem (3.6.1) to (3.6.8) if and only if u^* is a fixed point to G.

To allow us a greater amount of parallelism (as suggested later) we can uncouple Eqns (4.4.3) and (4.4.4) by considering two vectors $W^0_{1,k}(c)$ and $W^0_{2,k}(c)$, $k = 1, 2, \ldots, n$ and replacing (4.4.3) and (4.4.4) by

$$u'_{1,k}(c) = +\kappa[W^0_{1,k}(c) - W^0_{2,k}(c)]/[-\kappa R_{1,k}(c) + R_{2,k}(c)] \tag{4.4.7}$$

$$u'_{2,k}(c) = [W^0_{1,k}(c) - W^0_{2,k}(c)]/[-\kappa R_{1,k}(c) + R_{2,k}(c)] \tag{4.4.8}$$

Of course we have to update $W^0_{1,k}(c)$ and $W^0_{2,k}(c)$ using $W_{1,k}$ and $W_{2,k}$ computed from Eqn (4.4.2)

4.2. ITERATIVE ALGORITHMS

We describe now several iterative sequential and parallel algorithms which we have considered to solve the above problem. Notice first that (4.4.1) has the exact solution

$$R_i(r) = \frac{r\,\Delta\theta}{\sqrt{2}} \tanh\left(\frac{\sqrt{2}}{\Delta\theta} \ln \frac{r}{a_i}\right) \tag{4.4.9}$$

where $i = 1, 2$, $a_1 = a$ and $a_2 = b$.

4.2.1. Sequential Jacobi-type algorithm (J)

We start with the initial guess $u^0_{1,k} = A_{1,k}$, $u^0_{2,k} = A_{2,k}$ for $k = 1, 2, \ldots, n$. Then

For $p = 0, 1, 2, \ldots$ until convergence, do:

- For $k = 1, 2, \ldots, n$, do:
  1) use (4.4.2) to compute $W_{1,k}$ and $W_{2,k}$
  2) solve the system (4.4.3) and (4.4.4) to compute $[u^{p+1}_{1,k}(c)]'$ and $[u^{p+1}_{2,k}(c)]'$
  3) use (4.4.5) to compute $(u^{p+1}_{1,k})'$ and $(u^{p+1}_{2,k})'$
  4) use (4.4.6) to compute $u^{p+1}_{1,k}$ and $u^{p+1}_{2,k}$.

Of course it is understood that in (4.4.2) to (4.4.6), v_1 and v_2 (u_1 and u_2) are replaced by u^p_1 and u^p_2 (u^{p+1}_1 and u^{p+1}_2) respectively.

4.2.2. *Sequential Gauss-Seidel-type algorithm (GS)*

This is like the sequential Jacobi-type algorithm except that we make use of the updated vectors as soon as they become available. Of course we can introduce a relaxation parameter ω to obtain a SOR type algorithm.

4.2.3. *Modified sequential Jacobi-type algorithm (MJ)*

We start with the initial guess $u_{1,k}^0 = A_{1,k}$, $u_{2,k}^0 = A_{2,k}$, $W_{1,k}^0(c) = u_{1,k}^0(c)$, $W_{2,k}^0(c) = u_{2,k}^0(c)$, for $k = 1, 2, \ldots, n$. Then

For $p = 0, 1, 2, \ldots$ until convergence, do:

- For $k = 1, 2, \ldots, n$, do:
  1) use (4.4.2) to compute $W_{1,k}$
  2) use (4.4.7) to compute $[u_{1,k}^{p+1}(c)]'$
  3) use (4.4.5) to compute $(u_{1,k}^{p+1})'$
  4) use (4.4.6) to compute $u_{1,k}^{p+1}$
- For $k = 1, 2, \ldots, n$, do:
  1) use (4.4.2) to compute $W_{2,k}$
  2) use (4.4.8) to compute $[u_{2,k}^{p+1}(c)]'$
  3) use (4.4.5) to compute $(u_{2,k}^{p+1})'$
  4) use (4.4.6) to compute $u_{2,k}^{p+1}$

Of course $W_{1,k}^0(c)[W_{2,k}^0(c)]$ is to be updated using $W_{1,k}(c)[W_{2,k}(c)]$ computed from (4.4.2).

4.2.4. *Modified sequential Gauss-Seidel-type algorithm*

This is like the modified sequentia! Jacobi-type algorithm except that we make use of the updated components as soon as they become available.

The above methods are of sequential type. Now we give two algorithms of parallel type designed for multiprocessor.

4.2.5. *Asynchronous Jacobi-type algorithm (AJ)*

This is an asynchronous parallel version of the modified sequential Jacobi-type algorithm (MJ) defined in (4.2.3). We consider 2α processors P_j, $j = 1, 2, \ldots, 2\alpha$ and we assign to each one of them the evaluation of n_j components ($n_1 + n_2 + \cdots + n_{2\alpha} = 2n$). Each processor cyclically computes new values of each component in its subset using the values of all other necessary components (in or out of its subset) available at the beginning of a cycle, and releases all updated values at the end of each cycle.

To reduce the conflicts among adjacent processors and to simplify the simulation programming, we suppose that the processors are identical and that

$$n_1 = n_2 = \cdots = n_{2\alpha} = n/\alpha \qquad (4.4.10)$$

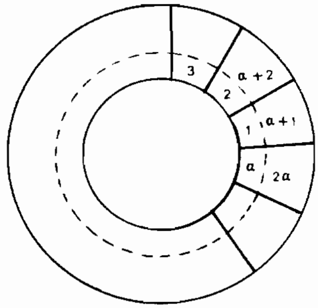

Fig. 3. Division of the domain among the 2α processors.

Therefore, each annulus is divided into n/α equal (in number of rays) sectors (Fig. 3), and each processor is now in communication with only three others (lower, right (left) and upper). For the case of two processors we have only side communications.

The 4 steps made by the monoprocessor in (4.2.3) to update one component are exactly carried out in parallel manner by each processor of the multiprocessor to update one component in its subset.

4.2.6. Asynchronous Gauss-Seidel-type algorithm (AGS)

This is an asynchronous parallel version of the modified sequential Gauss-Seidel-type algorithm (MGS) defined in (4.2.4). It is like the AJ algorithm except that each processor makes use of the updated components in its subset as soon as they become available.

It should be mentioned that the above algorithms are particular cases of the asynchronous algorithms for approximating a solution to a fixed point equation. The asynchronous algorithms were initially introduced by Chazan & Miranker (1969) under the name "chaotic relaxation" (also called delayed chaotic iterations). The terminology "asynchronous algorithms" was given by Baudet (1976, 1978) to chaotic relaxation with unbounded delays and he was the first to introduce and experiment on multiprocessor "C. mmp" algorithms like the AJ or AGS on linear algebraic system of equations. Many authors have contributed to the study of asynchronous algorithms and related algorithms. A good bibliography can be found in El-Tarazi (1981, 1982) and Spiteri (1984).

5. NUMERICAL RESULTS AND CONCLUSION

Consider the following heat conduction problem in composite media

$$\begin{cases} \dfrac{\partial^2 T_i}{\partial r^2} + \dfrac{1}{r}\dfrac{\partial T_i}{\partial r} + \dfrac{1}{r^2}\dfrac{\partial^2 T_i}{\partial \theta^2} = 0, & (r, \theta) \in \Omega_i \\ T_i(r, \theta) = \cos\theta + \beta_i, & (r, \theta) \in S_i \\ \begin{cases} T_1(r, \theta) = T_2(r, \theta) \\ \dfrac{\partial T_1}{\partial r}(r, \theta) = +\kappa \dfrac{\partial T_2}{\partial r}(r, \theta), \end{cases} & (r, \theta) \in S_3 \\ \text{for } i = 1, 2 \end{cases} \tag{5.1}$$

where

$$\begin{cases} \Omega_1 = \{(r, \theta), a \leqslant r \leqslant c, 0 \leqslant \theta \leqslant 2\pi\} \\ \Omega_2 = \{(r, \theta), c \leqslant r \leqslant b, 0 \leqslant \theta \leqslant 2\pi\} \\ S_1 = \{(r, \theta), r = a, 0 \leqslant \theta \leqslant 2\pi\} \\ S_2 = \{(r, \theta), r = b, 0 \leqslant \theta \leqslant 2\pi\} \\ S_3 = \{(r, \theta), r = c, 0 \leqslant \theta \leqslant 2\pi\}. \end{cases} \tag{5.2}$$

To treat this problem numerically, we have implemented the four sequential algorithms J, MJ, GS and MGS on monoprocessor, and simulated the parallel algorithms AJ and AGS for multiprocessors with 2, 4, 6, 8, 12, 18 and 24 processors. For the simulation schemes we refer to Julliand *et al.* (1980) and Spiteri (1984).

The numerical computations were made for $a = 0{\cdot}75$, $b = 1{\cdot}5$, $c = 1$, $\kappa = -0{\cdot}1$, $\beta_1 = 20$ and $\beta_2 = 5$. We have considered $n = 36$ rays. Each initial value problem was solved with the classical Runge Kutta method of order 4 and linear interpolation was used when necessary. The step size along each ray was taken as $\Delta r = 1/200$. This means taking 50(100) mesh points in the inner (outer) part of each ray, making a total number of mesh points equal to 36 (50 + 100) = 5400. Each iterative process was stopped when

$$\|u^{p+1} - u^p\| \leqslant 0{\cdot}5 \times 10^{-4} \tag{5.3}$$

Since the exact analytical solution of the above example is known to be

$$\begin{cases} T_i(r, \theta) = M_i \ln r + N_i + [P_i r + (Q_i/r)]\cos\theta \\ \text{for } i = 1, 2 \end{cases} \tag{5.4}$$

where

$$+\kappa M_2 = M_1 = -\kappa(\beta_1 - \beta_2)/\ln\left[\left(\frac{a}{c}\right)^{-\kappa}\frac{b}{c}\right] \tag{5.5.1}$$

$$N_1 = \beta_1 - M_1 \ln a, \; N_2 = \beta_2 - M_2 \ln b \tag{5.5.2}$$

$$\begin{cases} \delta = b(c^2 + a^2) - 2ac^2 - b\kappa(a^2 - c^2) \\ P_2 = \delta/[-\kappa(c^2 + b^2)(a^2 - c^2) - (c^2 - b^2)(c^2 + a^2)] \end{cases} \tag{5.5.3}$$

$$c^2 P_1 = 0{\cdot}5\{P_2[(c^2 - b^2) + \kappa(c^2 + b^2)] + (1 - \kappa)b\} \tag{5.5.4}$$

$$Q_1 = a(1 - aP_1), \; Q_2 = b(1 - bP_2) \tag{5.5.5}$$

we could therefore compute the errors of the considered algorithms; we found that each one gave numerical results correct to at least 3 decimal places. The sequential algorithms J(GS) and MJ(MGS) were almost equivalent.

The numerical results are given in Tables 1 and 2. The number appearing after each abbreviation designates the number of processors used. The inner (outer) relaxation is defined to be the set of the basic four actions described in (4.2.1) or (4.2.3) necessary to update one component. The speed-up is the ratio of the execution time of the sequential algorithm MJ(MGS) to the execution time of its parallel version

Table 1. Computational work of the modified sequential Jacobi-type algorithm MJ and the asynchronous Jacobi-type algorithms AJ2, AJ4, AJ6, AJ8, AJ12, AJ18 and AJ24.

Algorithm	MJ	AJ2	AJ4	AJ6	AJ8	AJ12	AJ18	AJ24
Inner relaxations	27 × 36	39 × 36	2 × 41 × 18	3 × 41 × 12	4 × 41 × 9	6 × 39 × 6	9 × 39 × 4	12 × 39 × 3
Outer relaxations	27 × 36	20 × 36	2 × 21 × 18	3 × 21 × 12	4 × 21 × 9	6 × 20 × 6	9 × 20 × 4	12 × 20 × 3
Speed-up		2·0	3·8	5·7	7·6	11·9	17·7	23·4
Efficiency %		100	95	95	95	99	98	98

Table 2. Computational work of the modified sequential Gauss-Seidel-type algorithm MGS and the asynchronous Gauss-Seidel-type algorithms AGS2, AGS4, AGS6, AGS8, AGS12, AGS18 and AGS24.

Algorithm	MGS	AGS2	AGS4	AGS6	AGS8	AGS12	AGS18	AGS24
Inner relaxations	16 × 36	32 × 36	2 × 31 × 18	3 × 31 × 12	4 × 31 × 9	6 × 29 × 6	9 × 31 × 4	12 × 31 × 3
Outer relaxations	16 × 36	16 × 36	2 × 16 × 18	3 × 16 × 12	4 × 16 × 9	6 × 15 × 6	9 × 16 × 4	12 × 16 × 3
Speed-up	...	1·5	3·0	4·5	5·9	9·4	13·1	17·3
Efficiency %		75	75	75	74	78	73	72

AJ(AGS). The efficiency is the ratio of the speed-up to the number of processors used for the parallel algorithm, and it is given as percentages.

The geometry of the domain (Fig. 3), its uniform partition by the processors and the fact that an outer relaxation costs almost twice as much as an inner relaxation are reflected in the numerical results; and they favoured the AJ algorithm in reducing the number of outer relaxations to 20 or 21 instead of 27 as in the MJ algorithm, thus giving an efficiency greater than 95%.

These numerical results of parallel computations, obtained by simulation, should only be considered to illustrate the behaviour of parallel algorithms. They show a clear advantage for asynchronous algorithms over synchronous classical sequential methods. We may note that many different types of parallelism can be considered.

Finally, we conclude by saying that the proposed approach combines the simplicity of the classical invariant imbedding associated with the by lines approximation and the efficiency of the asynchronous algorithms.

REFERENCES

Anwar, M.N. & El-Tarazi, M.N. 1985. Asynchronous algorithms for Poisson's equation with nonlinear boundary conditions. Computing **34**: 155–68.

Baudet, G.M. 1976. Asynchronous iterative methods for multiprocessors. Research Report of the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania, U.S.A.

Baudet, G.M. 1978. The design and analysis of algorithms for asynchronous multiprocessors. Ph.D. thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania, U.S.A.

Bellman, R. 1975. An introduction to invariant imbedding. Wiley Interscience, Canada.

Chazan, D. & Miranker, W.L. 1969. Chaotic relaxation. Linear Algebra and its Applications **2**: 199–222.

El-Tarazi, M.N. 1981. Contraction et ordre partiel pour l'etude d'algorithmes synchrones et asynchrones en analyse numérique. Thése de Doctorat ès Sciences. Université de Besançon, France.

El-Tarazi, M.N. 1982. Some convergence results for asynchronous algorithms. Numerische Mathematik **39**: 325–40.

Jaworski, A. 1981. Boundary elements for heat conduction in composite media. Applied Mathematical Modelling **5**: 45–48.

Julliand, J., Perrin, G.R. & Spiteri, P. 1980. Simulations d'exécutions parallèles d'algorithmes de relaxation asynchrone. Rapport de Recherche E.R.A. CNRS Micro-systèmes et Robotique No. 070906 et de Mathematiques No. 070654, Université de Besançon, France.

Meyer, G.H. 1973. Initial value methods for boundary value problems. Theory and application of invariant imbedding. Academic Press, New York.

Meyer, G.H. 1978. The method of lines for Poisson's equations with nonlinear or free boundary conditions. Numerische Mathematik **29**: 329–44.

Spiteri, P. 1984. Contribution a l'etude de grands systemes non lineaires: Comportement d'algorithmes itératifs, stabilité de systemes continus. Thése de Doctorat ès Sciences. Université de Besançon, France.

(Received 18 March 1987, revised 24 December 1988)

خوارزميات تسلسلية وأخرى لامتواقتة للتوصيل الحراري في وسط مؤلف

محمد نبيه الترزي ومحمد نعيم يحيى أنور*
قسم الرياضيات بجامعة الكويت ، ص . ب . ٥٩٦٩ ،
الصفاة ١٣٠٦٠ ، الكويت

خلاصة

في هذا البحث ، أمكن إيجاد حل للتوصيل الحراري ، في حالة الإستقرار ، خلال وسط مكون من مادتين مختلفتين متجانستين في المستوى ، باستخدام طريقة الخطوط المستقيمة . وفي هذه الطريقة ، تتحول المعادلات إلى نظام المعادلات التفاضلية ، من الرتبة الثانية ، ذات الشروط الحدية .

ويتحول هذا النظام ، بدوره ، إلى مسألة النقطة الثابتة وذلك باستخدام طريقة « التوطين الصامد » وبواسطة معادلات تفاضلية إعتيادية . ثم أمكن حل المعادلات الناتجة عدديا بطرق تسلسلية وأخرى لامتواقتة .

* العنوان الحالي : قسم الرياضيات الهندسية بكلية الهندسة ، جامعة الاسكندرية ، الحدرة ، الاسكندرية ، جمهورية مصر العربية .