ISSN: 0128-7680
Pertanika J. Sci. & Technol. 2(2): 149-158 (1994)

A Parallel AGE Method for Parabolic Problems with Special Geometries

Zaiton Muda, Mohamed Othman and Chuah Chin Yew
Department of Computer Science
Faculty of Science and Environmental Studies
Universiti Pertanian Malaysia
43400 Serdang, Selangor D.E., Malaysia

Received 3 February 1993

ABSTRAK

Kaedah tak tersirat kumpulan berselang-seli (TTKS) merupakan satu kaedah lelaran tak tersirat bagi masalah parabola yang melibatkan domain sekata telah dilaksanakan dalam sistem Sequent S27. Kaedah TTKS ini sesuai bagi komputer selari kerana ia mempunyai tugas-tugas yang terpisah dan merdeka, contohnya blok-blok (2 × 2) yang boleh dilaksanakan serentak tanpa melibatkan satu sama lain. Makalah ini menerangkan pembangunan dan perlaksanaan algoritma selari TTKS. Keputusan-keputusan yang diperolehi daripada perlaksanaan selari ini dibandingkan dengan perlaksanaan secara jujukan.

ABSTRACT

The alternating group explicit (AGE), an explicit iterative method for parabolic problems involving regular domains of cylindrical symmetry is implemented in parallel on a MIMD Sequent S27 system. The AGE method is suitable for parallel computers as it possesses separate and independent tasks, i.e (2 × 2) blocks which can be executed at the same time without interfering with each other. This paper reports the development and implementation of the parallel AGE algorithm. The results from parallel implementation are compared with those of the sequential implementation.

Keywords: parallel AGE, parallel computer, MIMD

INTRODUCTION

Let us consider the following parabolic equation in one-space dimension given by

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial r^2} + \frac{\alpha}{r}\frac{\partial U}{\partial r} \qquad (1.1)$$

subject to the initial-boundary conditions

$$U(r, 0) = f(r), \qquad 0 \le r \le 1$$

and

$$\frac{\partial U}{\partial U}(0, t) = 0,\ U(1, t) = 0,\ \text{for } 0 \le t \le T \qquad (1.2)$$

This equation reduces to the simple diffusion equation when $\alpha = 0$ and by putting $\alpha = 1$ and $\alpha = 2$ respectively, it becomes a parabolic problem with cylindrical and spherical symmetry. The application of the AGE algorithm to the heat equation (Evans and Sahimi 1987) is now extended for problem (1.1).

PROBLEM FORMULATION

A uniformly-spaced network whose mesh points are $r_i = i\Delta r$, $t_j = j\Delta t$ for $i = 0, 1, \ldots, m, m+1$ and $j = 0, 1, \ldots, n, n+1$ is used with $\Delta r = 1/(m+1)$, $\Delta t = T/(n+1)$ and mesh ratio $\lambda = \Delta t/(\Delta r)^2$.

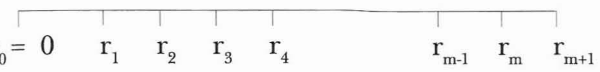

Fig. 1

A weighted approximation to (1.1) at the point (r_i, t_{j+1}) is thus given by Saul'yev (1964),

$$[1 + 2(1+\alpha)\lambda\theta]u_{0,j+1} - 2(1+\alpha)\lambda\theta u_{1,j+1} = [1 - 2(1+\alpha)\lambda(1-\theta)u_{0,j}$$

$$+ 2(1+\alpha)\lambda(1-\theta)u_{1,j} \qquad (2.1)$$

at the axis $r = 0$ and

$$-p_i\theta u_{i-1,j+1} + (1 + 2\lambda\theta)u_{1,j+1} - q_i\theta u_{i+1,j+1} = p_i(1-\theta)u_{i-1,j}$$

$$+[1 - 2\lambda(1-\theta)]u_{i,j} + q_i(1-\theta)u_{i+1,j} \qquad (2.2)$$

for $i = 1, 2, \ldots, m$ and points not on the axis, where

$$p_i = (1 - \alpha/2i)\lambda$$

and

$$q_i = (1 + \alpha/2i)\lambda.$$

Following Sahimi and Muda (1988), approximations (2.1) and (2.2) may be written as

$$\begin{bmatrix} a_0 & b_0 & & & () \\ c_1 & a_1 & b_1 & & \\ & c_2 & a_2 & b_2 & \\ & . & . & . & \\ & & . & . & . \\ & () & c_{m-1} & a_{m-1} & b_{m-1} \\ & & & c_{m-1} & a_m \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ . \\ . \\ u_{m-1} \\ u_m \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ . \\ . \\ f_{m-1} \\ f_m \end{bmatrix}$$

or

$$A\underset{\sim}{u} = \underset{\sim}{f} \qquad (2.3)$$

where

$$a_0 = 1 + 2(1+\alpha)\lambda\theta, \qquad a_1 = a = (1 + 2\lambda\theta), \text{ for } i = 1, 2, \ldots, m;$$

$$b_0 = -2(1+\alpha)\lambda\theta, \qquad b_i = -q_i\theta, \text{ for } i = 1, 2, \ldots, m;$$

$$c_i = -p_i\theta, \text{ for } i = 1, 2, \ldots, m;$$

$$f_0 = [1 - 2(1+\alpha)\lambda(1-\theta)]u_{0,j} + 2(1+\alpha)\lambda(1-\theta)]u_{1,j}$$

$$f_i = p_i(1-\theta)u_{i-1,j} + [1 - 2\lambda(1-\theta)]u_{i,j} + q_i(1-\theta)]u_{i+1,j'}$$

$$\text{for } i = 1, 2, \ldots, m-1;$$

$$f_m = p_m(1-\theta)u_{m-1,j} + [1 - 2\lambda(1-\theta)]u_{m,j} + q_m[\theta u_{m+1,j+1} + (1-\theta)]u_{m+1,j}].$$

Note that (2.3) corresponds to the fully implicit, the Crank-Nicolson and the classical explicit methods when $\theta = 1, 1/2$, and 0 respectively with $O((\Delta r)^2 + \Delta t)$, $O((\Delta r)^2 + (\Delta t)^2)$ and $O((\Delta r)^2 + \Delta t)$ accuracy.

Without loss of generality, assume that m is odd. Then, we have an even number $(m+1)$ of internal mesh points at which we seek the solutions of (2.3) along each time level. We now perform the following splitting of A:

$$A = G_1 + G_2 \tag{2.4}$$

where

$$G_1 = \left[\begin{array}{cc:cc:c:cc}
\frac{1}{2}a_0 & b_0 & & & & & \\
c_1 & \frac{1}{2}a & & & & \multicolumn{2}{c}{0} \\
\hdashline
 & & \frac{1}{2}a & b_2 & & & \\
 & & c_3 & \frac{1}{2}a & \cdot & & \\
\hdashline
 & & \cdot & & \cdot & & \\
\hdashline
 & & & & \cdot & \frac{1}{2}a & b_{m-1} \\
\multicolumn{2}{c:}{0} & & & & c_m & \frac{1}{2}a
\end{array}\right]_{(m+1)\times(m+1)} \tag{2.5}$$

$$G_2 = \left[\begin{array}{c|cc|cc|c|cc|c}
\frac12 a_0 & & & & & & & & \\
\hline
 & \frac12 a & b_1 & & & & \multicolumn{2}{c|}{()} & \\
 & c_2 & \frac12 a & & & & & & \\
\hline
 & & & \frac12 a & b_3 & & & & \\
 & & & c_4 & \frac12 a & \cdot & & & \\
\hline
 & & & & \cdot & & \cdot & & \\
 & & & & \cdot & & \cdot & & \\
\hline
 & () & & & & \cdot & \frac12 a & b_{m-2} & \\
 & & & & & & c_{m-1} & \frac12 a & \\
\hline
 & & & & & & & & \frac12 a
\end{array}\right]_{(m+1)\times(m+1)} \qquad (2.6)$$

Following Evans and Sahimi (1987), the following iterative AGE convergent scheme was derived,

$$(G_1 + \hat{r}I)\underset{\sim}{u}^{(k+1/2)} = (\hat{r}I - G_2)\underset{\sim}{u}^{(k)} + \underset{\sim}{f}$$

$$(G_2 + \hat{r}I)\underset{\sim}{u}^{(k+1)} = (G_2 - (1-\omega)\hat{r}I)\underset{\sim}{u}^{(k)}$$

$$+ (2-\omega)\,\hat{r}\underset{\sim}{u}^{(k+1/2)} \qquad (2.7)$$

for any $0 \leq \omega \leq 2$ and $\hat{r} > 0$ being a fixed acceleration parameter along each intermediate (half-time) level or iterate. A particular choice of $\omega = 0$ gives us the Peaceman-Rachford (PR) scheme and for $\omega = 1$, we obtain the variant due to Douglas and Rachford (DR). Both stable variants are known to have truncation errors of the order $T_{PR} = O\ ((\Delta r)^2 + (\Delta t)^2)$ and $T_{DR}\ ((\Delta r)^2 + \Delta t)$ respectively.

Following Sahimi and Muda (1988) we arrive at the computation of the solution of our geometrical problem:

1) at level (k + 1/2)

$$u_0^{(k+1/2)} = (A_0 u_0^{(k)} + B_0 u_1^{(k)} + C_0 u_2^{(k)} + D_0)/\alpha_0$$

$$u_1^{(k+1/2)} = (A_1 u_0^{(k)} + B_1 u_1^{(k)} + C_1 u_2^{(k)} + D_1)/\alpha_0$$

$$u_i^{(k+1/2)} = (A_i u_{i-1}^{(k)} + Bu_1^{(k)} + C_i u_{i+1}^{(k)} + D_i u_{1+2}^{(k)} + E_i)/\alpha_{i/2}$$

$$\text{for } i = 2, 4, 6, \ldots, m-1 \qquad (2.8a)$$

$$u_{i+1}^{(k+1/2)} = (\bar{A}_i u_{i-1}^{(k)} + B_i u_i^{(k)} + Bu_{i+1}^{(k)} + C_i u_{1+2}^{(k)} + \bar{D}_i)/\alpha_{i/2}$$

$$\text{for } i = 2, 4, 6, \ldots, m-1 \qquad (2.8b)$$

where

$$A_0 = \bar{s}_0 s_0, \quad B_0 = -b_0 s, \quad C_0 = b_0 b_1, \quad D_0 = \bar{s}_0 f_0 - f_1$$

$$A_1 = -c_1 s_0, \quad B_1 = -\bar{s}_0 s, \quad C_1 = -\bar{s}_0 b_1, \quad D_1 = c_1 f_0 - f_1 + \bar{s}_0 f_1$$

$$\bar{A}_i = -c_i c_{i+1}, \quad B_i = -s c_{i+1}, \quad C_i = -s b_{i+1}, \quad \bar{D}_i = \bar{s} f_{i+1} - c_{i+1} f_i$$

$$A_i = -c_i \bar{s}, \quad B_1 = s\bar{s}, \quad C_i = -b_i s, \quad D_i = b_i b_{i+1}, \; E_i = s f_i - i b_i$$

with

$$\bar{s}_0 = \hat{r} + a_0/2, \quad \bar{s} = \hat{r} + a/2, \quad s_0 = \hat{r} - a_0/2, \text{ and } s = \hat{r} - a/2.$$

2) at level (k + 1)

$$u_0^{(k+1)} = (q_0 u_0^{(k)} + d_0 u_0^{(k+1/2)}/\bar{s}_0$$

$$u_i^{(k+1)} = (P_i u_i^{(k)} + Q_i u_{i+1}^{(k)} + R u_i^{(k+1/2)} + S_i u_{i+1}^{(k+1/2)}/\hat{\alpha}_{(i+1)/2}$$

$$\text{for } i = 1, 3, 5, \ldots, m-2 \tag{2.9a}$$

$$u_{i+1}^{(k+1)} = (\bar{P}_i u_i^{(k)} + P_i u_{i+1}^{(k)} + \bar{Q}_i u_i^{(k+1/2))} + R u_{i+1}^{(k)}/\hat{\alpha}_{(i+1)/2}$$

$$\text{for } i = 1, 3, 5, \ldots, m-2 \tag{2.9b}$$

where

$$P_i = \bar{s}q - b_i c_{i+1}, \quad Q_i = b_i(\bar{s} - q), \quad R = \bar{s}d, \quad S_i = -b_i d$$

$$P_i = c_{i+1}(\bar{s} - q) \quad Q_i = -c_{i+1} d$$

with

$$q_0 = a_0/2 - (1 - \omega)\hat{r}, \quad q = a/2 - (1 - \omega)\hat{r} \text{ and } d = (2 - \omega)\hat{r}.$$

Since the equations 2.8a & 2.8b and 2.9a & 2.9b are explicit, then their solution on a parallel computer is possible.

PARALLEL AGE EXPLOITATION

A parallel algorithm has been developed and implemented to solve the one-space dimension parabolic equation on a MIMD shared memory parallel computer.

The mesh of points (*Fig. 1*) is decomposed into a subset of points, each of which is assigned to a processor. As we have seen, the computation of the solution of our geometrical problem involves iterations of the two sweeps.

For the first sweep of each mesh point, each computational molecule of equations 2.8a & 2.8b is again assigned to a processor. The computational molecules are then solved depth by depth in parallel in bottom-up order (*Fig. 2(a)*). This method is also known as the balanced binary tree method. The depth of such a tree will be bound by (log n) and the complexity of such an algorithm will be O (log n) where n is the number of nodes. The maximum number of processors employed in this discipline is n/2.

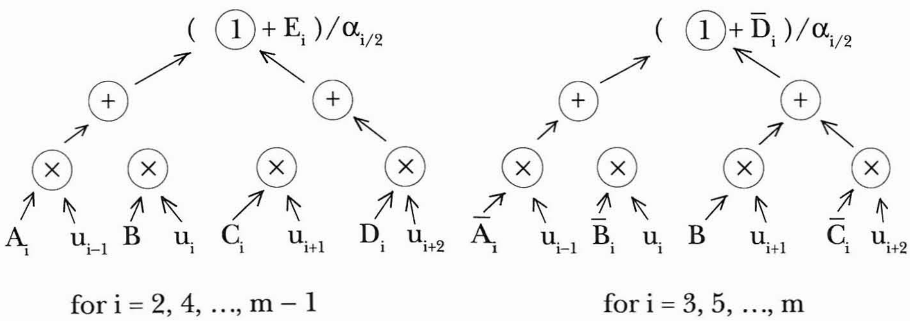

Fig. 2(a)

The second sweep is started after the first sweep has been completed. The computational molecules of equation 2.9a & 2.9b are also solved using the same technique (*Fig. 2(b)*). Then a test of convergence is carried out after the second sweep. Further iterations are needed until a prescribed tolerance ε is achieved.

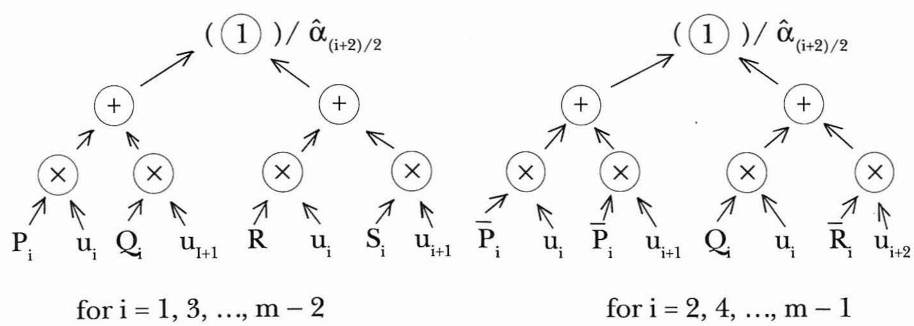

Fig. 2(b)

The algorithm for the parallel AGE method is then described as follows.

Algorithm begin
for h = 1 to n do
    begin
        $T = ht$
        $k = 0$
        while (**not converge**) and ($k < MAX$)
            begin
                $k=k+1$
                $u_0^{(k+1/2)} = \ldots$
                $u_1^{(k+1/2)} = \ldots$
                for i = 2 to m−1 **in parallel** do
                    begin
                        for j = 4 to 7 **in parallel** do
                            $A_{i,j} = A_{i,2j} * A_{i,2j+1}$
                        for j = 2 to 3 **in parallel** do
                            $A_{i,j} = A_{i,2j} + A_{i,2j+1}$
                        $u_i^{(k+1/2)} = (A_{i,2} + A_{i,3} + A_{i,4})/\alpha_i$
                    end
            $u_i^{(k+1)} = \ldots$
            for i = 1 to m−2 **in parallel** do
        begin
                        for j = 4 to 7 **in parallel** do
                            $B_{i,j} = B_{i,2} * B_{i,2j+1}$
                        for j = 2 to 3 **in parallel** do
                            $B_{i,j} = B_{i,2j} + B_{i,2j+1}$
                    $u_i^{(k+1)} = (B_{i,2} + B_{i,3}) * /\alpha_i$
                end
            Test Convergence
            [abs($u_i^{(k+1)} - u_i^{(k)}$) < ε for all i]
            Replace $u_i^{(k+1)}$ with new $u_i^{(k)}$ for all i
        end
    end
Algorithm end.

As the method of divide-and-conquer (Evans and Sutti 1988) is widely applicable in sequential computation, a sequential program is developed and implemented to give its performance in contrast with the parallel.

EXPERIMENTAL RESULTS

Let us consider the cylindrical problem (Mitchell and Pearce 1963)

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial r^2} + \frac{1}{r}\frac{\partial U}{\partial r}, \qquad (0 \le r \le 1)$$

$$U(r, 0) = J_0(\beta_r),$$

$$\frac{\partial U}{\partial U}(0, t) = U(1, t) = 0, \qquad t > 0$$

where J_0 (βr) is the Bessel function of the first kind of order 0 and β is the first root of $J_0(\beta) = 0$. The exact solution is $U(r, t) = J_0(\beta r)e^{-\beta 2t}$.

This problem is implemented in parallel as well as in sequential on the sequent symmetry S27 system using the strategy discussed above. The programs are written in C language; compiled with the Symmetry C Compiler (version 6.2). The Sequent computer runs on the DYNIX operating system, a version of UNIX 4.2bsd that also supports most utilities, libraries and system calls provided by UNIX System V. Here the Sequent computer supports multi-tasking on 2 processors.

The accuracy of these parallel AGE method results has been verified (*Fig. 3(a)*) with the implicit-sequential results obtained from Sahimi and Muda (1988).

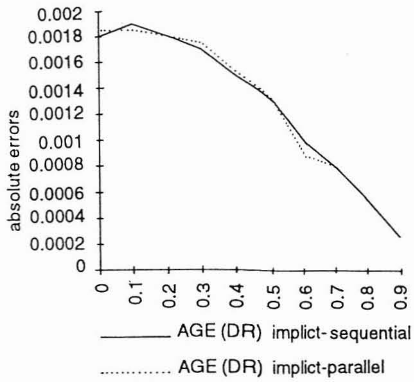

Fig. 3(a). The absolute errors of the numerical solutions to the cylindrical problems (where $\lambda = 1.0$, $t = 1.0$, $\Delta r = 0.1$, $\Delta t = 0.01$, $\hat{r} = 0.9$, $\omega = 1$)

In the implementations, the execution time of the two sweeps is obtained with the number of mesh points being increased (Table 1). By comparing the results of the implementations, we notice that the processing times for the parallel strategy are less than those of the sequential strategy (*Fig. 3 (b) & 3(c)*). This is mainly due to the effectiveness of the algorithm which enables a high percentage of the problem to be parallelized (in equations 2.8a, 2.8b, 2.9a and 2.9b).

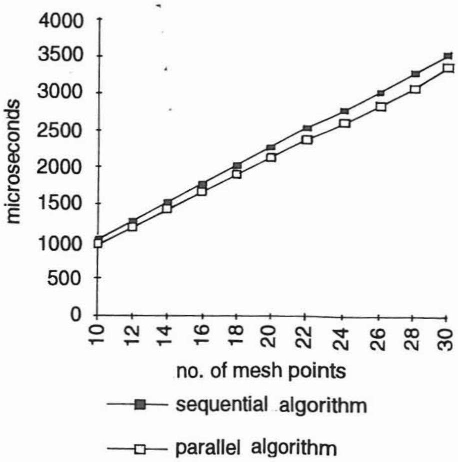

Fig. 3(b). The time execution performance of the first sweep

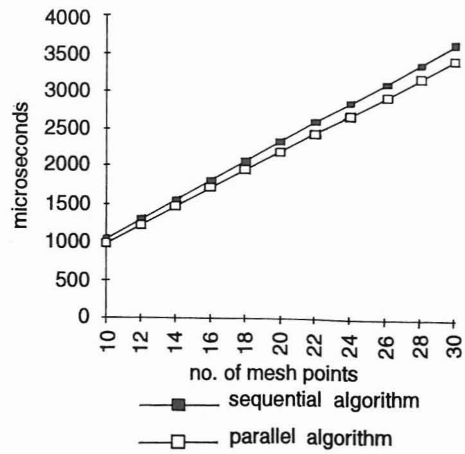

Fig. 3(c). The time execution performance of the second sweep

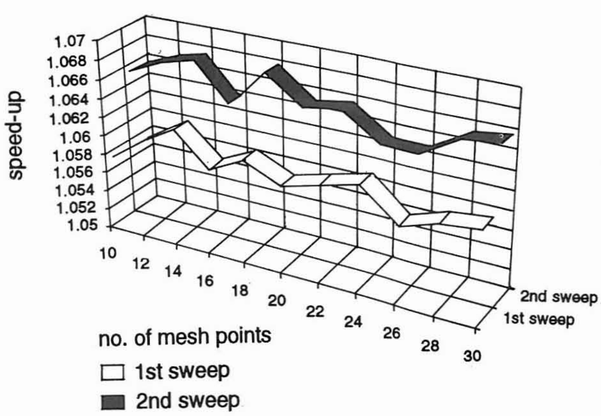

Fig. 3(c). Comparison of the speed-up of the parallel AGE method

TABLE 1
The execution time (in microseconds) and speed-up performance of the parallel AGE method (where λ = 1.0, t = 1.0, Δr = 0.1, Δt = 0.01, $\hat{r}$ = 0.9, ω = 1)

no. of mesh points	sequential algorithm		parallel algorithm		parallel speed-up	
	1st sweep	2nd sweep	1st sweep	2nd sweep	1st sweep	2nd sweep
10	1035	1012	979	950	1.057201	1.065263
12	1293	1264	1220	1185	1.059836	1.066667
14	1558	1516	1467	1420	1.062031	1.067606
16	1822	1767	1721	1661	1.058687	1.063817
18	2065	2024	1947	1895	1.060606	1.068074
20	2323	2274	2194	2135	1.058797	1.065105
22	2589	2529	2443	2373	1.059763	1.066574
24	2841	2782	2678	2617	1.060866	1.063049
26	3095	3028	2926	2846	1.057758	1.063949
28	3353	3279	3166	3078	1.059065	1.065302
30	3613	3595	3410	3373	1.059531	1.065904

In *Fig. 3(d)*, the speed-up for the first sweep is less than the second. This is because the second sweep has a higher percentage of parallelism than the first. Due to the limitation factor of hardware facilities (that is the number of processors is two), further conclusions on speed-up for more than two processors cannot be made. However, we expect a better speed-up as we increase the number of processors. This will mean more subproblems can be solved simultaneously.

REFERENCES

EVANS, D. J. and M.S. SAHIMI. 1987. Alternating group explicit iterative method (AGE) to solve parabolic and hyperbolic partial differential equations. In *Ann. Rev. Num. Fluid. Mech. & Heat Trans.* ed. Chong Lin Tien, 2nd ed. Hemisphere Pub. Co.

EVANS, D.J. and C. SUTTI. 1988. *Parallel Computing Methods, Algorithms and Applications.* Adam Hilger.

MITCHELL, A.R. and R.P. PEARCE. 1963. Explicit difference methods for solving the cylindrical heat conduction equation. *Maths. Comp.* **17**: 426-432.

PEACEMAN, D.W. and H.H. RACHFORD. 1955. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Maths.* **3**: 28-41.

SAHIMI, M.S. and Z. MUDA. 1988. An iterative explicit method for parabolic problems with special geometries. *UKM Tech. Report.* PPK/Mac 88/LT 7.

SAUL'YEV, V. K. 1984. *Integration of Equations of Parabolic Type by the Method of Nets.* Oxford: Pergamon Press.