Parallel Algorithms On Some Numerical Techniques Using PVM Platform On A Cluster Of Workstations

Norma Alias, Dr.Mohd Salleh Sahimi Mohamed, Prof.Abdul Rahman Abdullah
Jabatan Matematik, Fakulti Sains, UTM, Skudai, Johor.
Jabatan Matematik dan Sains, Universiti Tenaga Nasional, Kajang, SEL.
Jabatan Komputeran Industri, FTSM, UKM, Bangi SEL.
norm_ally@hotmail.com, Sallehs@uniten.edu.my, ara@ftsm.ukm.my

Abstract

In this paper, a few parallel algorithms are explained in solving one dimensional heat model problem using Parallel Virtual Machine (PVM). This research focuses on two iteration methods, Iterative Alternating Decomposition Explicit Method (IADE) and Alternating Group Explicit Scheme (AGE). Conjugate Gradient (CG) is selected as an alternative method to accelerate the convergent and efficiency of these two iteration methods.

keywords

Parallel Virtual Machine(PVM), Iterative Alternating Decomposition Explicit Method(IADE), Alternating Group Explicit Scheme(AGE), Conjugate Gradient(CG).

1.0 Introduction

Six strategies of parallel algorithms are implemented to exploit the convergence of IADE_CG. In the domain decomposition strategy Ω, the IADE_CG Michell-Fairweather which is fully explicit, is derived to produce the approximation of grid-i and not totally dependent on the grid $(i-1)$ and $(i+1)$. In IADE_CG Red Black and IADE_CG SOR strategies, the domain Ω is decomposed into two different subdomains Ω^H and Ω^M. The concept of multidomain is observed in the IADE_CG Multicoloring method. The decomposition of domain Ω into w different groups of domain. The Domains for colors $1, 2, 3, ..., w$ are noted as $\Omega^{w_1}, \Omega^{w_2}, ..., \Omega^{w_w}$. For the computational grid for domain Ω, its execution started with level Ω^{w_1}, followed by level Ω^{w_2} and ends with level Ω^{w_w}. On the vector iteration strategy, parallel IADE_CG is run in two sections. This method converges if the inner convergence criterion is achieved for each section.

On the strategy of Incomplete Block LU preconditioners on slightly overlapping subdomains, the domain Ω is decomposed into p processors with incomplete subdomain $\overline{\Omega}$. This strategy used a preconditioners, the incomplete factorization with certain parameters of algebraic boundary condition. Thus, AGE_CG algorithm is shown to be extremely straightforward as implemented in parallel algorithms using PVM. CG is chosen as the alternative parallel algorithm because it does not increase the communication time between the processors. The application of CG is a correction to the parallel and sequential algorithms of IADE_CG and AGE_CG.

2.0 IADE_CG Algorithms

IADE with Mitchell-Fairtweather is introduced by Evans and Sahimi (1992) for solving the one dimensional heat problems. A generalized finite different approximation to the different equation at point $(x_i, t_{j+\frac{1}{2}})$ is given by,

$$\begin{aligned} -\lambda\theta u_{i-1,j+1} + (1+2\lambda\theta)u_{i,j+1} - \lambda\theta u_{i+1,j+1} &= \lambda(1-\theta)u_{i-1,j} + [1-2\lambda(1-\theta)]u_{i,j} \\ &+ \lambda(1-\theta)u_{i+1,j}, \qquad i = 1, 2, 3, ..., m, \end{aligned} \tag{1}$$

which leads to the three-point formulae, displayed in matrix form as

$$\mathbf{Au} = \mathbf{f} \tag{2}$$

The IADE iterative employs the fractional splitting strategy,

$$\begin{aligned}
(rI + \mathbf{G}_1)u^{(k+\frac{1}{2})} &= (rI - g\mathbf{G}_2)u^{(k)} + \mathbf{f} && (3)\\
(rI + \mathbf{G}_2)u^{(k+1)} &= (rI - g\mathbf{G}_1)u^{(k+\frac{1}{2})} + g\mathbf{f} && (4)
\end{aligned}$$

where the coefficient matric A can be decomposed into the matrices $\mathbf{G}_1$ and $\mathbf{G}_2$, as

$$\mathbf{A} = \mathbf{G}_1 + \mathbf{G}_2 - \frac{\mathbf{G}_1\mathbf{G}_2}{g}$$

The consistent $g = \frac{6+r}{6}$, r is acceleration parameter and the constituent matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ take the bidiagonal form (lower and upper respectively)
i. at the $(k+\frac{1}{2})^{th}$ iterate

$$u_i^{(k+\frac{1}{2})} = \frac{1}{d}(-k_{i-1}u_{i-1}^{(k+\frac{1}{2})} + q_i u_i^{(k)} + w u_{i+1}^{(k)} + f_i), \quad i = 1, 2, ..., m \quad , \quad k_0 = 0 \tag{5}$$

with $q_i = \frac{r-ge_i}{d}, \quad \forall i \in [1, m]$
ii. at the $(k+1)^{th}$ iterate

$$\begin{aligned}
u_{m+1-i}^{(k+1)} = \frac{1}{d_{m+1-i}}(-v_{m-i}u_{m-i}^{(k+\frac{1}{2})} + s u_{m+1-i}^{(k+\frac{1}{2})} + g f_{m+1-i} - h_{m+1-i}u_{m+2-i}^{(k+1)}),\\
i = 1, 2, ..., m, \quad v_0 = 0
\end{aligned} \tag{6}$$

with $d_i = r + e_i$ and $\forall i \in [1, m]$. Sequential IADE shows that the approximation solution for $u_i^{(k+\frac{1}{2})}$ is dependent on $u_{i-1}^{(k+\frac{1}{2})}$ and approximation solution for $u_{m+1-i}^{(k+1)}$ is dependent on $u_{m+2-i}^{(k+1)}$). The parallel algorithm strategies are implemented to avoid the sequential IADE situation.

3.0 AGE_CG Algorithms

Through the Alternating Direction Implicit (ADI), AGE methods with Peaceman-Rachford variation is created to be more extremely powerful, flexible and it offers users many advantages . The accuracy of this method is comparable if not better than that of the GE class of problems as well as other existing schemes (Evans and Abdullah, 1983). This method employs the fractional splitting strategy and the implicit form is as follows,

$$\begin{aligned}
u^{(k+\frac{1}{2})} &= (\mathbf{G}_1 + rI)^{-1}[(rI - \mathbf{G}_2)u^{(k)} + \mathbf{f}]\\
u^{(k+1)} &= (\mathbf{G}_2 + rI)^{-1}[(rI - \mathbf{G}_1)u^{(k+\frac{1}{2})} + \mathbf{f}] \ ,
\end{aligned} \tag{7}$$

we have

$$\mathbf{A} = \mathbf{G}_1 + \mathbf{G}_2$$

If we assume m to be odd then $\widehat{\mathbf{G}}$ could be written as,

$$\widehat{\mathbf{G}} = \begin{bmatrix} r_2 & b \\ c & r_2 \end{bmatrix}_{(2\times 2)}$$

where, $r_2 = r + \frac{a}{2}$. The alternating implicit nature of the (2×2) groups where the implicit and explicit values are given on the forward and backward levels for sweeps on the $(k+\frac{1}{2})^{th}$ and $(k+1)^{th}$ levels, with $r_1 = r - \frac{a}{2}$, $r_2 = r + \frac{a}{2}$ end $\Delta = r_2^2 - bc$
i. at the $(k+\frac{1}{2})^{th}$ iterate

$$\begin{aligned} u_1^{(k+\frac{1}{2})} &= \frac{r_1 u_1^{(k)} - b u_2^{(k)} + f_1}{r_2} \\ u_i^{(k+\frac{1}{2})} &= \frac{A u_{i-1}^{(k)} + B u_i^{(k)} + C u_{i+1}^{(k)} + D u_{i+2}^{(k)} + E_i}{\Delta} \\ u_{i+1}^{(k+\frac{1}{2})} &= \frac{\widetilde{A} u_{i-1}^{(k)} + \widetilde{B} u_i^{(k)} + \widetilde{C} u_{i+1}^{(k)} + \widetilde{D} u_{i+2}^{(k)} + \widetilde{E}_i}{\Delta} \end{aligned} \tag{8}$$

with $i = 2, 4, 6, ..., m-1$, $A = -cr_2$, $B = r_1 r_2$, $C = -br_1$, $E_i = r_2 f_i - b f_{i+1}$,

$D = \{^{o, i=m-1}_{b^2, i \neq m-1}$
and $\widetilde{A} = -cr_2$, $\widetilde{B} = r_1 r_2$, $\widetilde{C} = -br_1$, $\widetilde{E}_i = r_2 f_i - b f_{i+1}$,

$\widetilde{D} = \{^{o, i=m-1}_{-br_2, i \neq m-1}$

ii. at the $(k+1)^{th}$ iterate

$$\begin{aligned} u_i^{(k+1)} &= \frac{P u_{i-1}^{(k+\frac{1}{2})} + Q u_i^{(k+\frac{1}{2})} + R u_{i+1}^{(k+\frac{1}{2})} + S u_{i+2}^{(k+\frac{1}{2})} + T_i}{\Delta} \\ u_{i+1}^{(k+1)} &= \frac{\widetilde{P} u_{i-1}^{(k)} + \widetilde{Q} u_i^{(k)} + \widetilde{R} u_{i+1}^{(k)} + \widetilde{S} u_{i+2}^{(k)} + \widetilde{T}_i}{\Delta} \\ u_m^{(k+1)} &= \frac{-c u_{m-1}^{(k+\frac{1}{2})} + r_1 u_m^{(k+\frac{1}{2})} + f_m}{r_2} \end{aligned} \tag{9}$$

with $Q = r_1 r_2$, $R = -br_1$, $S = b^2$, $T_i = r_2 f_i - b f_{i+1}$,
$\widetilde{P} = \{^{o, i=1}_{c^2, i \neq 1}$
and $\widetilde{Q} = -cr_1$, $\widetilde{R} = \widetilde{Q} = r_1 r_2$, $S = -br_2$, $\widetilde{T}_i = -cf_i + r_2 f_{i+1}$,
All the equations are not dependent on every point $i-1$, i and $i+1$ event for every time step. This is the advantages to create the parallel algorithm for AGE.

Furthermore, Conjugate Gradient algorithm (CG) is exploited to accelerate the convergence and efficiency of of IADE and AGE iterations. If $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ is residual vector at level k and if $\mathbf{A}$ is symmetric matrix $(m \times m)$ end positively defined, the CG algorithm is as follows,

$$\begin{array}{rl} \text{i.} & \mathbf{p}_k \leftarrow \mathbf{A}\mathbf{d}_k \\ \text{ii.} & \mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{d}_k, \qquad \alpha_k = \dfrac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{d}_k^T \mathbf{p}_k} \\ \text{iii.} & \mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k \mathbf{p}_k, \\ \text{iv.} & \mathbf{d}_{k+1} \leftarrow \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k, \qquad \beta_k = \dfrac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \end{array} \tag{10}$$

if $\mathbf{x}_0$ is initial value $\mathbf{x}$ and $\mathbf{d}_0 = \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is initial value $\mathbf{r}$ and $\mathbf{d}$. A few type of CG algorithms is created by chosen the value of acceleration parameter α_k and β_k. The chosen normal value of acceleration parameter [Reid(1971)] is,

$$\alpha_k = \frac{\mathbf{d}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{d}_k} \qquad end \qquad \beta_k = \frac{-\mathbf{p}_k^T \mathbf{r}_{k+1}}{\mathbf{p}_k^T \mathbf{d}_k}$$

The first step, the IADE_CG Algorithms is executed by using the equation (5) and sweeps along the domain Ω. The second step, executes CG algorithms by using the equation (10)$(i - iv)$. These two steps are continued until a specified convergence criterion is satisfied, when the requirement $|u_{i,j}^{(k-1)} - u_{i,j}^{(k)}| \le \varepsilon$ is met and ε becomes the convergence criterion. The accuracy of the solution at these grid points was determined by computing its root means square error. For an introduction to conjugate gradient acceleration of iterative methods, see Hageman and Young (1981).

4.0 Parallel IADE_CG Algorithms

1. IADE_CG Michell-Fairweather

The IADE_CG Michell-Fairweather which is fully explicit is derived to produce the approximation of grid-i and which is not totally dependent on the grid $(i-1)$ and $(i+1)$. The approximation at the first and second intermediate levels are computed directly by inverting $(rI + \mathbf{G}_1)$ and $(rI + \mathbf{G}_2)$. The explicit form of equation (3) and (4) is given by

$$u^{(k+\frac{1}{2})} = (rI + \mathbf{G}_1)^{-1}(rI - g\mathbf{G}_2)u^{(k)} + \mathbf{f} \tag{11}$$

$$u^{(k+1)} = (rI + \mathbf{G}_2)^{-1}(rI - g\mathbf{G}_1)u^{(k+\frac{1}{2})} + g\mathbf{f} \tag{12}$$

In the first step, IADE_CG Michell-Fairweather implements the equations (11) and (12) and the second step, exploits the CG algorithm as in equation (10)$(i - iv)$

The parallel IADE_CG Michell-Fairweather is executed by decomposing the m grid into $\frac{m}{p}$ groups of grid and assigning the groups into p processors. For example if $G_1, G_2, ..., G_{\frac{m}{p}}$ are the groups of grid and $p_1, p_2, ..., p_p$ are the processors, the $G_i, i = 1, 2, ..., \frac{m}{p}$ are assigned to $p_j, j = 1, 2, ..., p$.

2. IADE_CG Red Black

In IADE_CG Red Black strategy, the domain Ω is decomposed into two different subdomains Ω^H and Ω^M. Ω^H is the approximate solution on the odd grids and Ω^M is the approximate solution on the even grids. Computation on Ω^H is executed followed by Ω^M. These two subdomains are not dependent on each other. Ω^H is decomposed into groups, $H_1, H_2, ..., H_{\frac{m}{p}}$ and Ω^M is decomposed into groups, $M_1, M_2, ..., M_{\frac{m}{p}}$. Every group of H_i and M_i, $i = 1, 2, ..., \frac{m}{p}$ is assigned to processors p . IADE_CG Red Black is run in parallel for each subdomain in alternating way on time steps $(k + \frac{1}{2})$ and $(k+1)$. The parallel IADE_CG Red Black formulae for (3) and (4) are as follows,

i. at the $(k + \frac{1}{2})^{th}$ iterate

$$\begin{aligned} a.u_i^{k+\frac{1}{2}} &= -\frac{q_i u_i^k}{d} + \frac{\omega}{d}(-k_{i-1}u_{i-1}^k + q_i u_i^k + w_i u_{i+1}^k + f_i), \\ i &= 1, 3, 5, ..., m \\ b.u_i^{k+\frac{1}{2}} &= -\frac{q_i u_i^k}{d} + \frac{\omega}{d}(-k_{i-1}u_{i-1}^{k+\frac{1}{2}} + q_i u_i^k + w_i u_{i+1}^k + f_i), \\ i &= 2, 4, 6, ..., m-1, k_0 = w_0 = 0 \end{aligned}$$

ii. at the $(k+1)^{th}$ iterate

$$\begin{aligned}
a.u_{m+1-i}^{k+1} &= -\frac{su_{m+1-i}^{k+\frac{1}{2}}}{d_{m+1-i}} + \frac{\omega}{d_{m+1-i}}(-v_{m-i}u_{m-i}^{k+\frac{1}{2}} + su_{m+1-i}^{k+\frac{1}{2}} + gf_{m+1-i} - h_{m+1-i}u_{m+2-i}^{k}), \\
i &= 1,3,5,...,m, v_0 = 0 \\
b.u_{m+1-i}^{k+1} &= -\frac{su_{m+1-i}^{k+\frac{1}{2}}}{d_{m+1-i}} + \frac{\omega}{d_{m+1-i}}(-v_{m-i}u_{m-i}^{k+\frac{1}{2}} + su_{m+1-i}^{k+\frac{1}{2}} + gf_{m+1-i} - h_{m+1-i}u_{m+2-i}^{k+1}), \\
i &= 2,4,6,...,m-1
\end{aligned}$$

and acceleration parameter $\omega = 1$.
The next step, exploits the CG algorithm as in equation $(10)(i - iv)$

3. IADE_CG RBSOR
Using the well-known fact of the IADE_CG Red Black, the parallel algorithm for IADE_CG RBSOR takes the form similar to IADE_CG Red Black but the acceleration parameter ω was chosen to provide the most rapid convergence.

4. IADE_CG Multicoloring
By the definition of multidomain, domain Ω is decomposed into w different groups. IADE_CG Multicoloring is an advanced concept of IADE_CG Red Black. If $w = 2$, then IADE_CG Multicoloring is equaled to IADE_CG Red Black.

The Domains for colors $1,2,3,...,w$ are noted as $\Omega^{w_1}, \Omega^{w_2}, ..., \Omega^{w_w}$. The subdomain Ω^{w_i} is distributed into different groups of grid $W_{i1}, W_{i2}, ..., W_{i\frac{m}{wp}}$, where $i = 1,2,...,w$. In the process of assignment, $W_{ij}, i = 1,2,...,w$ and $j = 1,2,..., \frac{m}{wp}$ are mapped into the processors p in the alternating way.

At each time step, the computational grid for domain Ω started its execution with level Ω^{w_1}, followed by level Ω^{w_2} and ends with level Ω^{w_w}. The next step, exploits the CG algorithm as in equation $(10)(i - iv)$

5. IADE_CG vector
On the vector iteration strategy, the parallel IADE_CG is run in two convergence sections. The first section is at the $(k + \frac{1}{2})$ time step and the second section is the $(k+1)$ time step. This method converges if the inner convergence criterion is achieved for each section. The inner convergence criterions $\varepsilon^{(k+\frac{1}{2})}$ and $\varepsilon^{(k+1)}$ are definite global convergence criterion ε.

The CG algorithm as in equation $(10)(i - iv)$ is attained before the examination of global convergence criterion.

6. IADE_CG Incomplete Block LU
On the strategy of Incomplete Block LU preconditioners on slightly overlapping subdomains, the domain Ω is decomposed into p processors with incomplete subdomain $\overline{\Omega}$. This strategy used the incomplete factorization with parameter β and δ of algebraic boundary condition as follows,
i. at the $(k + \frac{1}{2})^{th}$ iterate

$$k_{i-2}u_{i-2}^{Q} + \beta k_{i-2}u_{i-1}^{Q} - \delta k_{i-2}u_{i}^{Q} + (1+\beta)k_{i-2}u_{i-1}^{Q} + \delta k_{i-2}u_{i+1}^{Q} = f_{i-1}$$

ii. at the $(k+1)^{th}$ iterate

$$d_{i-1}u_{i-1}^{R} + h_{i-1}u_{i}^{R} = f_{i-1}$$

The CG algorithm as in equation (10) $(i - iv)$ is attained after the execution of IADE_CG Incomplete Block LU.

5.0 Numerical experiments

The problem statement under consideration is based on one dimensional heat conductor equation.(Smith (1979))

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad 0 \leq x \leq 1, \quad 0 < t, \tag{13}$$

subject to the initial condition, $U(x, 0) = sin(\pi, x), \quad 0 \leq x \leq 1$
and the boundary condition, $U(0, t) = U(1, t) = 0, \quad 0 < t \leq 1$

6.0 Experiment Result And Parallel Performances

This paper presents the numerical properties of the parallel solver on the homogeneous architecture of 12 PC systems with Linux operating, Intel Pentium IV processors, speedup 20GB HDD and connected with internal network Intel 10/100 NIC. The following definitions are used to measure the parallel algorithms in term of speedup,efficiency and effectiveness.

$$\text{Speedup ratio} S_p = \frac{T_1}{T_p} \tag{14}$$

$$\text{Efficiency} C_p = \frac{S_p}{p} \tag{15}$$

$$\text{Effectiveness} F_p = \frac{S_p}{C_p} \tag{16}$$

where $C_p = pT_p$, T_1 is the execution time on a serial machine and T_p is the computing time on a parallel machine with p processors. Gauss Seidel Red Black is chosen as the control scheme. The number of iterations, the $rmse$, the maximum $rmse$, the maximum error and the execution time cost in sequential algorithms are shown in Figure 1. As shown in the figure 1 , the sequential IADE_CG, AGE_CG and IADE_CG Michell-Fairweather take the same number of relaxations to reach the convergence criterion ε with the minimum execution time. With domain decomposition, the implementation of parallel strategies, IADE_CG Incomplete block LU, IADE_CG Red Black, IADE_CG Red Black SOR, IADE_CG Multicoloring and IADE_CG vector are increased in the number of relaxations and the execution time. The convergence rates, the $rmse$, the maximum $rmse$ and the maximum error of all the sequential algorithms are slightly similar.

Figure 2 shows the execution time vs. number of processors. As expected, the computation time decreases with the increasing p. The 3 categories of the parallel algorithms indicated the best execution time in the following arrangement,

1. AGE_CG and IADE_CG Incomplete Block,
2. IADE_CG Red Black SOR and IADE_CG Red Black,
3. IADE_CG Multicoloring and IADE_CG Vector

Using this control scheme , IADE_CG Michell-Fairweather and IADE_CG vector are not suitable to be implemented as the parallel algorithms scheme for IADE_CG . It involves high computational cost and communication cost. All the parallel algorithms for these categories have a better execution time compared to their sequential algorithms as the number of processors are increased.

Figure 3 shows the speedup plotted against the number of processors p. As observed from the experiment, a nice speedup can be obtained for all applications with 12 processors except for IADE_CG Michell-Fairweather. The reductions in execution time often becomes smaller when a large number of processors is used. This phenomenon as stated in Amdahl's law, indicates that the number of processors increases, the communication cost (e.g., the latency for massage passing) and the cost for global operations will eventually become dominant over local computation cost after a certain stage.

The IADE_CG Multicoloring, IADE_CG Red Black and IADE_CG Red Black SOR are good in terms of speedup and efficiency where data decomposition is run asynchronously and concurrently at every time step. AGE_CG and IADE_CG Incomplete Block allow for inconsistencies due to load balancing when the extra computation cost is needed for boundary condition. From equation (14) to equation (16),

$$F_p = \frac{S_p}{pT_p} = \frac{E_p}{T_p} = \frac{E_p S_p}{T_1}$$

which shows that F_p measure both speedup and efficiency. Therefore, a parallel algorithm is said to be effective when it maximizes F_p hence, $F_p T_1 (= S_p E_p)$. The optimal performance of the effectiveness could be obtained when more than 12 processors are used. Figure 4 shows the best effectiveness in the following order,
1. IADE_CG Incomplete Block,
2. AGE_CG
3. IADE_CG Red Black
4. IADE_CG Red Black SOR
5. IADE_CG Vector
6. IADE_CG Multicoloring

7.0 Conclusion

Conjugate Gradient (CG) can be regarded as a very efficient technique to increase the convergent and efficiency of these two iteration methods. The stable and high accuracy of IADE_CG Incomplete Block is found to be an alternative parallel algorithm for IADE_CG on PVM. All the parallel algorithms maintained the accuracy of the sequential IADE_CG algorithms. Parallel AGE_CG is inherently explicit, the domain decomposition strategy is efficiently utilized, straightforward to implement on PVM. These parallel algorithms is available to be implemented on heterogeneous architecture cluster of workstations. The experiment proved that the communication and computing times affect the speedup ratio, efficiency and effectiveness for the number of processors p. Therefore, we reach the conclusion that these parallel algorithms gets better efficiencies when it is implemented on a cluster workstations for solving a large-scale problems.

8.0 Acknowledgment

We wish to express our gratitude and indebtedness to the Universiti Kebangsaan Malaysia, Universiti Tenaga Nasional and Malaysian Government for providing the moral and financial support under IRPA grant for the successful completion of this project.

9.0 References

Al Geist, Al., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. & Sunderam, V. 1994. *PVM : Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing.* Cambridge: MIT Press.

Chalmer, A. & Tidmus, J. 1996. *Practical Parallel Processing - An Introduction to Problem Solving in Parallel.* International Thomson Computer Press.

Evans, D. J. & Sahimi, M. S. 1989. The alternating Group Explicit Iterative method (AGE) to Solve Parabolic and Hyperbolic Partial Differential Equation. *In Annual Review of Numerical Fluid Mechanics and Heat Transfer.* **2**: 283–389.

Evans, D. J. & Sahimi, M. S. 1989. The alternating Group Explicit (AGE) Iterative method for Solving Parabolic Equations II: 3 Space Dimensional Problems. *International Journal Computer Mathematic* **26**: 117–142.

Evans, D. J. & Sahimi, M. S. 1988. The alternating Group Explicit (AGE) Iterative method for Solving Parabolic Equations I. *International Journal Computer Mathematic* **24**: 127–145.

Foster, I. 1996. *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering.* Inc: Addison- Weslley Publishing Company.

Hageman, L. A. & Young, D. M. 1981. *Applied Iterative Methods.* Academic, New York.

Lewis, T.G. & El-Rewini, H. 1998. *Distributed and Parallel Computing.* New York: Manning Publication.

Michael, J. Quinn. 1994. *Parallel Computing Theory and Practice.* London: Mc Graw-Hill.

Wilkinson, B. & Allen M. 1999.*Parallel Programming: Techniques and Applications Using Networked workstationss and Parallel Computers.* New Jersey : Prentice Hall.

Zamoya, A. Y.1996. *Parallel and Distribution Computing Handbook.* New York: Mc Graw-Hill.

method	AGE	IADE	LU	RBSOR	RB	MULTI	MF	VECTOR	GSRB
me (ms)	49.743	40.2641	50.4327	124.5471	126.1464	205.9478	1896.65	261.0082	156.4432
erations	250	252	305	362	363	362	252	252	600
rmse	1.5921E-09	1.5921E-18	1.5921E-09	1.5921E-16	1.5921E-09	1.5921E-09	1.5921E-18	1.5921E-09	1.5921E-09
ax_error	1.1102E-16	3.3307E-16	4.4409E-16	2.2204E-16	3.7921E-16	4.2306E-16	3.3307E-16	3.3307E-16	1.11E-16
qr_error	1.9846E-07	1.9793E-07	1.9846E-07	1.9846E-07	1.9846E-07	1.9846E-07	1.9793E-07	1.9846E-07	1.9846E-07
ve_rmse	5.3374E-17	5.3374E-17	5.3374E-17	5.3374E-17	5.3374E-17	5.3374E-17	5.3374E-17	5.3374E-17	5.3374E-17
m	720003	720003	720003	720003	720003	720003	720003	720003	720003
dx	1.3889E-06	1.3889E-06	1.3889E-06	1.3889E-06	1.3889E-06	1.3889E-06	1.3889E-06	1.3889E-06	1.3889E-06
dt	9.6450E-13	9.6450E-13	9.6450E-13	9.6450E-13	9.6450E-13	9.6450E-13	9.6450E-13	9.6450E-13	9.6450E-13
tt	4.8225E-11	4.8225E-11	4.8225E-11	4.8225E-11	4.8225E-11	4.8225E-11	4.8225E-11	4.8225E-11	4.8225E-11
lamda	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
teta	1	1	1	1	1	1	1	1	1
round	50	50	50	50	50	50	50	50	50
r	0.8	0.8	1.1	1.3	1.29	0.98	0.8	0.9	-
ry	-	-	-	1.1	1	0.92	-	-	-
rz	-	-	-	1	1	1.09	-	-	-
tol	1.0000E-15	1.0000E-15	1.0000E-15	1.0000E-15	1.0000E-15	1.0000E-15	1.0000E-15	1.0000E-15	1.0000E-15
tol_y	-	-	-	-	-	-	-	1.0000E-15	-
tol_z	-	-	-	-	-	-	-	1.0000E-15	-
								1038+890	

Figure 1 Sequential algorithms for IADE

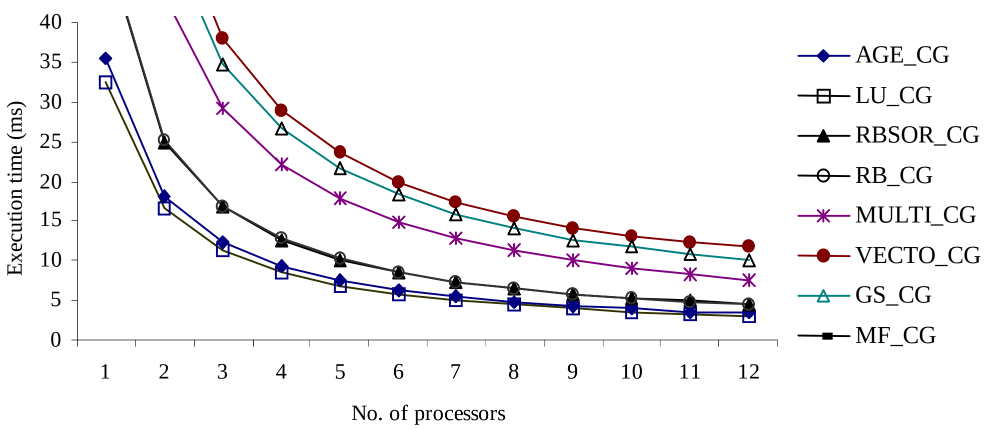

Figure 2 The execution time vs. number of processors

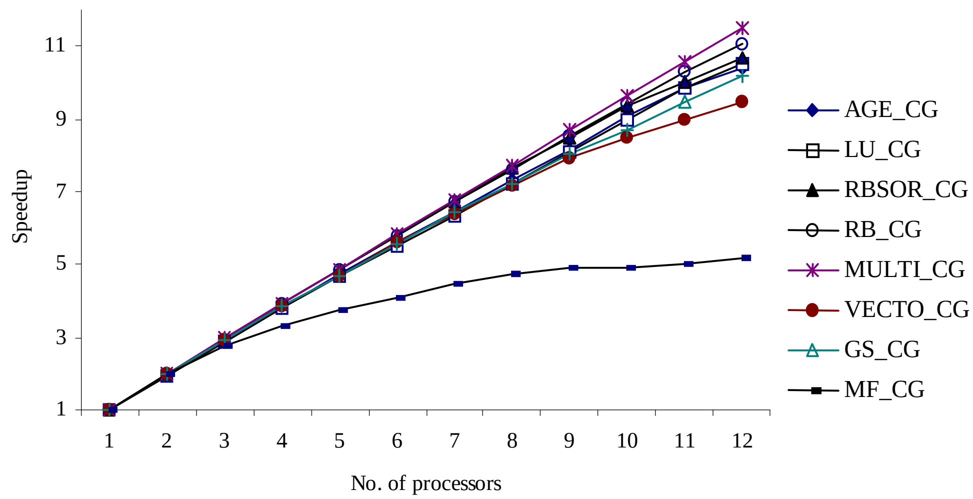

Figure 3 The speedup vs. number of processors

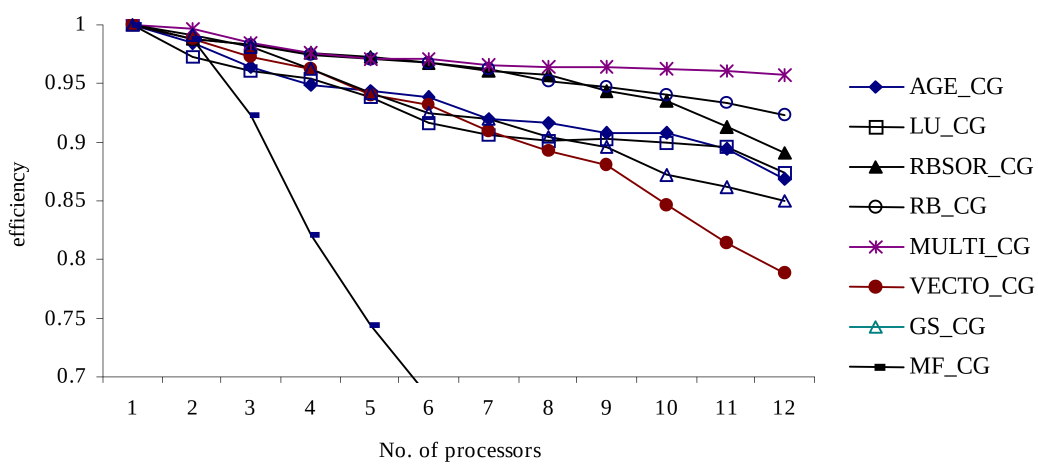

Figure 4 The efficiency vs. number of processors

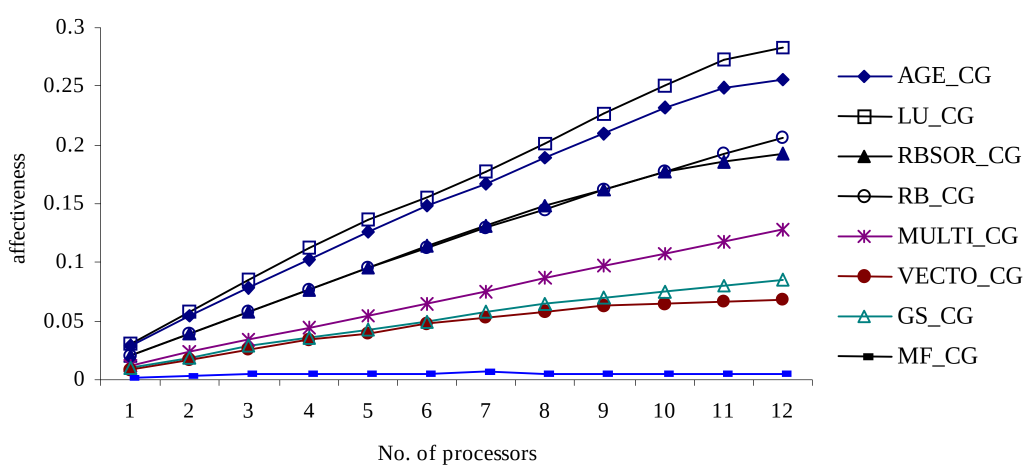

Figure 5 The effectiveness vs. number of processors