Jurnal Teknologi Maklumat & Multimedia 1(2004): 53-61

A Parallel New High Order Iterative Algorithm on Shared Memory Multiprocessors Parallel Computer

MOHAMED OTHMAN & ABDUL RAHMAN ABDULLAH

ABSTRACT

A new fast high-order points iterative algorithm of $O(h^4)$ applied to linear systems arising from discretization of 2D Poisson problem was recently introduced by the writers. This algorithm shows drastic reduction in execution time as compared to the standard high-order points iterative algorithm. In this paper, the parallel implementation of the algorithm with optimal strategy on shared memory multiprocessors (SMP) was presented and discussed. The numerical results of the test problem are included.

ABSTRAK

Satu algoritma lelaran titik bertertib tinggi baru dan terpantas $O(h^4)$ yang diaplikasikan kepada sistem linear hasil daripada pengdiskretan masalah Poisson 2D telah diperkenalkan oleh penulis. Algoritma ini telah menunjukkan penurunan masa pelaksanaan yang drastik berbandingkan dengan algoritma lelaran titik bertertib tinggi piawai. Dalam makalah ini, implementasi algoritma selari tersebut dengan strategi optima pada multipemproses ingatan berkongsi (SMP) dibentangkan dan dibincangkan. Hasil berangka daripada masalah ujian akan disertakan.

INTRODUCTION

The parallel high order iterative algorithm which incorporates the standard high order scheme which is also known as compact high order scheme for solving a large and sparse linear system has been implemented successfully by many researchers. One of the most outstanding parallel algorithm that uses the scheme was proposed by Spotz, et al. (1998). Theoretically, the standard high order scheme of $O(h^4)$ was derived by Coltaz (1960). Based on the scheme, several experiments were carried out and the results obtained have shown that it has higher accuracy, see Gupta (1984). Othman et al. (2001) derived a new scheme, which is known as a new fast high order scheme for solving the 2D Poisson equation. From the experimental results, they found that the new scheme is shown to have drastic improvement in execution time as compared to the standard high order scheme.

Derivation of a New High Order Scheme

Let us consider the 2D Poisson equation as our model problem, which can be represented mathematically as:

$$u_{xx} + u_{yy} = f(x,y), \qquad (x,y) \in \Omega^h, \tag{1}$$

subject to the Dirichlet boundary conditions and satisfying the exact solution, $u(x,y) = g(x,y)$ for $(x,y) \in \Omega^h$. The discretization resulted in a large and sparse linear system. Hence, the iterative method is considered as suitable approach for solving such a linear system.

Consider Equation (1) on a unit square, Ωh with the grid spacing h in both directions, $x_i = x_0 + ih$ and $y_i = y_0 + ih$ for all $i,j = 0,1,\ldots,n$. Assume that $u_{xxyy} = u_{yyxx}$ due to the continuity of $u(x,y)$ on Ω^h.

Based on the cross orientation approximation and central difference formula, the displacements i and j which correspond with Δx and Δy respectively changes to $\sqrt{2}h$. Equation (2) can be approximated at any points (x_i,y_j) using the finite difference formula and yields:

$$u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1} + 4u_{i,j} \cong 2h^2 (u_{xx} + u_{yy})_{i,j} + \frac{h^4}{6}(u_{xxxx} + 6u_{xxyy} + u_{yyyy})_{i,j} + O(h^6). \tag{2}$$

Equation (2) is known as a rotated five points stencil of $O(h^2)$ provided the second and third terms on the right-hand side are ignored. Since the accuracy of the stencil is not good, it is possible to derive the higher order of accuracy. Again the finite difference formula is used to derive the high order approximation. By taking width $2h$, approximation to Equation (1) at the point (x_i,y_j) takes the form,

$$u_{i+2,j} + u_{i-2,j} + u_{i,j-2} + u_{i,j+2} + 4u_{i,j} \cong 4h^2(u_{xx} + u_{yy})_{i,j} + \frac{4h^4}{3}(u_{xxxx} + u_{yyyy})_{i,j} + O(h^6). \tag{3}$$

Multiplying Equation (2) by 4 and adding it with Equation (3), we have:

$$u_{i+2,j} + u_{i-2,j} + u_{i,j-2} + u_{i,j+2} + 4(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) - 20u_{i,j} \cong 12h^2(u_{xx} + u_{yy})_{i,j} + 2h^4(u_{xxxx} + u_{yyyy})_{i,j}\; 4h^4(u_{xxyy})_{i,j} + O(h^6). \tag{4}$$

Double derivatives of Equation (1) with respect to x and y, we obtain:

$$(u_{xxxx} + u_{xxyy})_{i,j} = (f_{xx})_{i,j}, \tag{5}$$

and

$$(u_{xxyy} + u_{yyyy})_{i,j} = (f_{yy})_{i,j}, \tag{6}$$

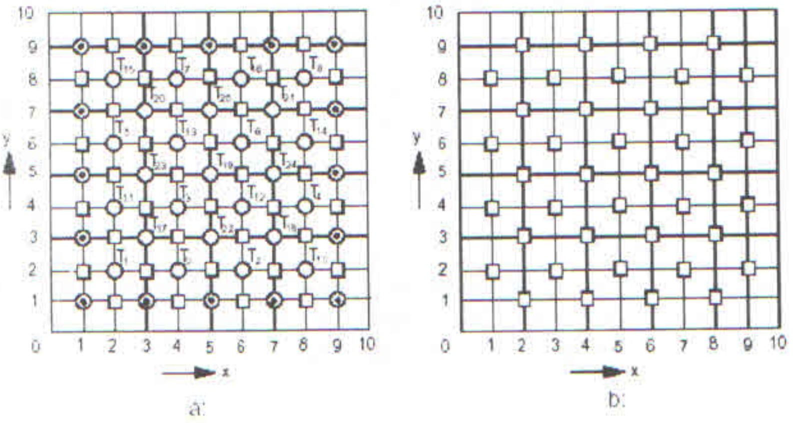
respectively. Multiply both Equations (5) and (6) by $2h^4$ and add, we can write Equation (4) as,

$$\begin{aligned} &u_{i+2,j} + u_{i-2,j} + u_{i,j-2} + u_{i,j+2} + 4(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) \\ &- 20u_{i,j} \cong 12h^2 f_{i,j} + 2h^4(f_{xx} + f_{yy})_{i,j} + O(h^6). \end{aligned} \tag{7}$$

For higher order approximation, replace second term on the right-hand side of Equation (7) by $h^2(f_{i+1,j+1} + f_{i-1,j+1} + f_{i+1,j-1} + f_{i-1,j-1} - 4f_{i,j})$ and ignore the third term, we obtain:

$$\begin{aligned} &u_{i+2,j} + u_{i-2,j} + u_{i,j-2} + u_{i,j+2} + 4(u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i-1,j-1}) \\ &- 20u_{i,j} \cong F_{i,j}, \end{aligned} \tag{8}$$

where $F_{i,j} = (8f_{i,j} + f_{i+1,j+1} + f_{i-1,j+1} + f_{i+1,j-1} + f_{i-1,j-1} - 4f_{i,j})$. Equation (8) is called a new high order scheme and its accuracy is $O(h^4)$. Details of the scheme and their computational molecule can be obtained in Othman et al. (2001). The compact high order scheme and their derivation were shown in details in Collatz (1960), Gupta (1984) and Sportz (1998).



FIGURES 1. (a) and (b) show the 4C ordering strategy as indicated $T_1, T_2, \ldots, T_{25}$ and the remaining mesh points in Ω^h, respectively for the parallel new high order algorithm with $n = 10$.

Assume the Ω^h is large with $n = 2(i+1)$ for any integer $i=1,2,\ldots$. Several ordering strategies of parallelizing all the points iterative algorithms have been studied and investigated (see Abdullah *et al.* 1996). However, only the optimal strategy is described in the following section.

A PARALLEL NEW HIGH ORDER ALGORITHM

Let the Ω^h be discretized and labeled into three different types of mesh points, $\odot$, o and $\square$ (see Figure 1a). All the o points (or tasks T_i) are allocated to the available processors in four colors (4C) strategy, white (w), yellow (y), green (g) and red (r). Note that all points of type $\odot$ will be executed first in parallel by using the rotated five point scheme with the natural strategy.

Applying this strategy to Equation (8) in turn with such strategy to each task T_i leads to the following linear system:

$$\begin{bmatrix} D_w & E & F & J \\ E_T & D_y & F & J \\ F_T & F^T & D_g & G \\ J_T & J^T & G^T & D_r \end{bmatrix} \begin{bmatrix} u_w \\ u_y \\ u_g \\ u_r \end{bmatrix} = \begin{bmatrix} f_w \\ f_y \\ f_g \\ f_r \end{bmatrix}, \tag{9}$$

where D_i the blocks D_i are diagonal and hence invertible. Thus, the S.O.R relaxation technique to Equation (9) will result to:

$$\left.\begin{aligned} u_w^{(k+1)} &= \xi u_w^{(k)} + \omega_e D_w^{-1}\left(f_w - Eu_y^{(k)} - Fu_g^{(k)} - Ju_r^{(k)}\right), \\ u_y^{(k+1)} &= \xi u_y^{(k)} + \omega_e D_y^{-1}\left(f_y - E^T u_w^{(k+1)} - Fu_g^{(k)} - Ju_r^{(k)}\right) \\ u_g^{(k+1)} &= \xi u_r^{(k)} + \omega_e D_r^{-1}\left(f_r - J^T\left(u_w^{(k+1)} + u_g^{(k+1)}\right) - Gu_r^{(k)}\right) \\ u_r^{(k+1)} &= \xi u_r^{(k)} + \omega_e D_r^{-1}\left(f_r - J^T\left(u_w^{(k+1)} + u_y^{(k+1)}\right) - G^T u_g^{(k)}\right) \end{aligned}\right\}, \tag{10}$$

where $\xi = (1 - \omega_r)$ and is the acceleration factor. Since the evaluation of each task T_i within each group is independent of one another, we can evaluate Equation (10) in parallel in the following order:

$$u_w^{(k+1)} \mapsto u_y^{(k+1)} \mapsto u_g^{(k+1)} \mapsto u_r^{(k+1)}.$$

In other words, each iteration is split into four sweeps in parallel separated by a synchronizing call. This to ensure the updates in k^{th} sweep are completed before the updates in the $(k+1)^{th}$ sweep begin. After an iteration is completed, a local convergence check will be made by each processor, followed by a global convergence check. The iteration is terminated only if local convergence is achieved, which indicates that each processor has achieved local convergence. After the global convergence is attainted, the solution of the remaining mesh points i.e. points of type □ will be evaluated directly in parallel using the standard five points formula by assigning each row to a different processor at a time.

Results and Performance Evaluations

All the methods were applied to the following model of problem which was defined in unit Ω^h and described as $u_{xx} + u_{yy} = (x^2 + y^2)\, e^{xy}$. The problem is subjected to the Dirichlet boundary condition and satisfying the exact solution $u(x,y) = e^{xy}, (x,y) \subset \delta\Omega^h$ as shown in Figure 2.

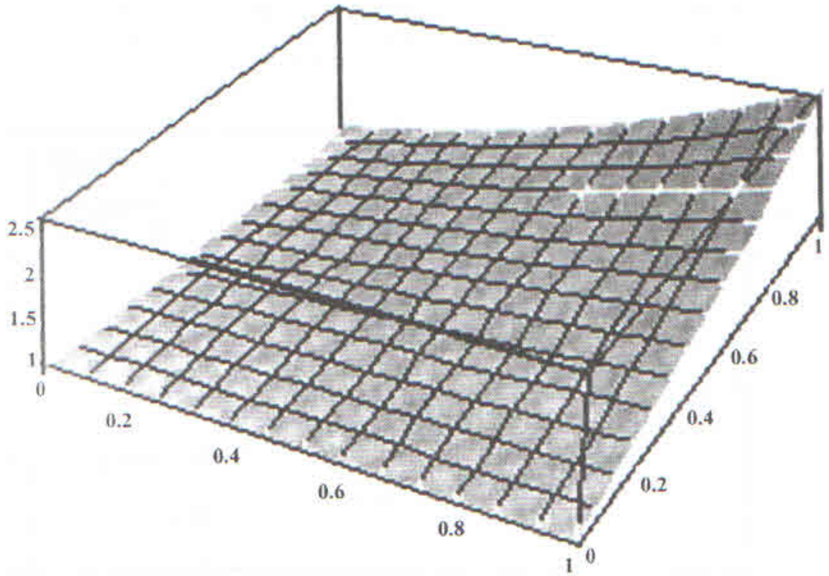


FIGURE 2. The exact solution of the problem in a unit Ω^h

Throughout the experiment, a tolerance of the $\varepsilon = 10^{-10}$ in the local convergence test was used. The experimental values of ω_e were obtained within ±0.01 by running the program for different values of ω_e and choosing the one(s) that gave the minimum number of iterations. The experiments were carried out on SMP parallel computer with several mesh sizes n as 36, 50, 70 and 100.

Table 1 shows the optimum value of ω_e, number of iterations, strategies and maximum errors. Table 2 shows the execution time and speedup for all the parallel high order iterative algorithms. The execution time, efficiency and temporal performance of all the parallel points iterative algorithms were plotted in Figures 3, 4 and 5, respectively.

TABLE 1. Acceleration factor ω_e, number of iteration, strategy and maximum error of all the parallel high order iterative algorithms

n	Algorithm	Acc. Factor, ω_e	No. of Iteration	Strategy	Max. Errors
36	Standard	1.84	146	4C	4.55×10^{-9}
	New	1.77	106	4C	1.33×10^{-6}
50	Standard	1.88	201	4C	1.24×10^{-9}
	New	1.83	160	4C	3.24×10^{-7}
70	Standard	1.91	276	4C	4.39×10^{-10}
	New	1.87	235	4C	8.85×10^{-8}
100	Standard	1.94	406	4C	1.12×10^{-10}
	New	1.90	359	4C	2.19×10^{-8}

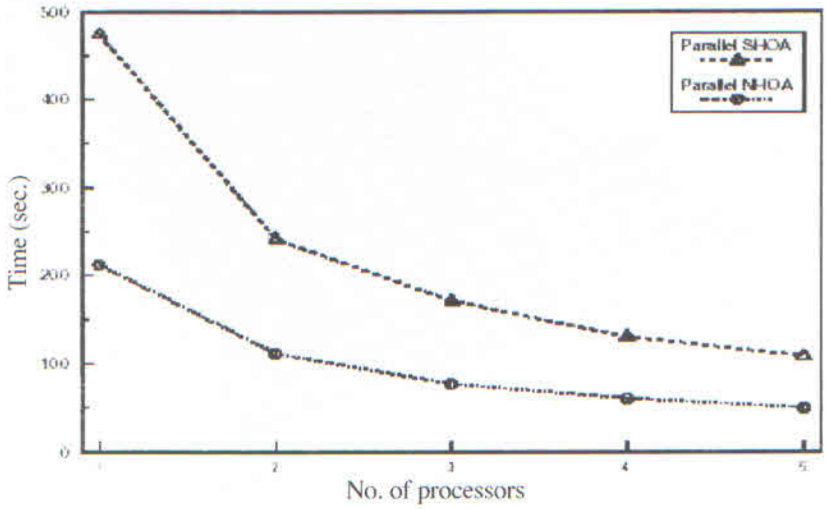


FIGURE 3. The execution time vs. no. of processors of all the parallel high order iterative algorithms n=100

TABLE 2. The numerical results of all the parallel high order iterative algorithms

n	No. of Processors	Parallel High Order Algorithms			
		Standard		New	
		Time	Speedup	Time	Speedup
	1	20.8174	1.0000	7.6300	1.0000
	2	11.3899	1.8277	4.2367	1.8009
36	3	8.3684	2.4876	3.1235	2.4427
	4	6.4995	3.2029	2.4777	3.0794
	5	5.3320	3.9042	2.0345	3.7502
	1	55.8694	1.0000	9.5020	1.0000
	2	29.9873	1.8631	10.6912	1.8421
50	3	20.7415	2.6736	7.6046	2.5645
	4	16.7213	3.3412	5.9125	3.2984
	5	13.6320	4.0984	4.8893	3.9887
	1	156.6552	1.0000	65.9847	1.0000
	2	82.6981	1.8943	35.0348	1.8834
70	3	56.1729	2.7888	24.8417	2.6562
	4	43.7951	3.5770	18.8630	3.4981
	5	36.5112	4.2906	15.8445	4.1045
	1	473.8377	1.0000	212.3990	1.0000
	2	241.7539	1.9600	111.4662	1.9055
100	3	171.1779	2.7681	76.8834	2.7626
	4	130.0251	3.6442	60.6213	3.5037
	5	108.0489	4.3854	49.6224	4.2803

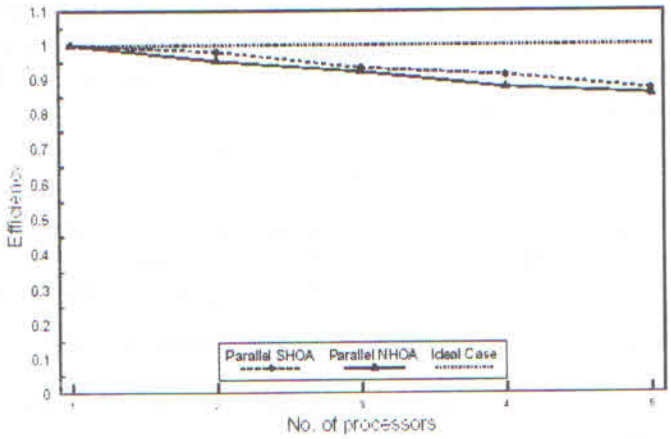


FIGURE 4. An Efficiency vs. No. of Processors of All the Parallel High Order Iterative Algorithms n=100

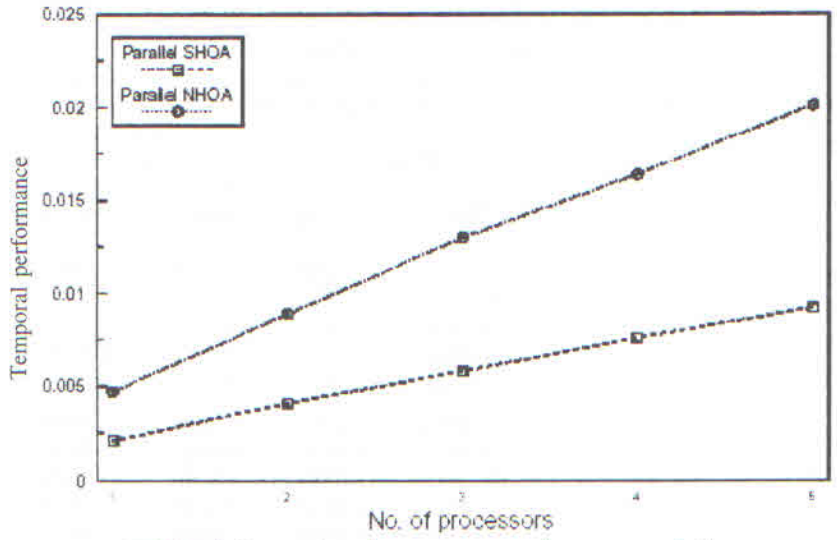


FIGURE 5. Temporal performance vs. no. of processors of all the parallel high order iterative algorithms n=100

CONCLUSION

The results obtained show that the parallel new high order algorithm with 4C strategy is faster than the parallel standard high order algorithm for any number of processors (Tables 1 and 2). It is also indicated in the graphs of execution time, efficiency, temporal performance versus no. of processors as stated in Figures 3, 4, 5, respectively. This is due to the fact that the mesh points involved in the iteration process are less than the other algorithm.
It can be concluded that the parallel new high order algorithm is the most superior and effective method among the two algorithms particularly for solving 2D Poisson problem. In the future, the algorithm will be implemented on the SMP cluster architecture.

REFERENCES

Abdullah, A. R. and Ali, N. H. M. 1996. Comparative Study of Parallel Strategies for the Solution of Elliptic PDEs, *Parallel Algorithm and Applications* 10: 93-103.

Collatz, L. 1960. *The Numerical Treatment of Differential Equations*, Berlin: Springer-Verlag.

Gupta. M. M. 1984. A Fourth Order Poisson Solver, *Journal Computational Physic* 55: 166-172.

Othman, M. and Abdullah, A. R. 2001. A Fast Higher Order Poisson Solver, *Sains Malaysiana* 30: 77-86.

Spotz, W. F. and Carey G. F. 1998. Iterative and Parallel Performance of High-Order Compact System, *SIAM Journal of Scientific Computing* 19(1): 1-14.

Mohamed Othman
Fakulti Sains Kompter & Teknologi Maklumat
Universiti Putra Malaysia
43400 UPM Serdang
Selangor D.E.
mothman@fsktm.upm.edu.my

Abdul Rahman Abdullah
Fakulti Teknologi & Sains Maklumat
Universiti Kebangsaan Malaysia
43400 UKM Bangi
Selangor D.E.
ara@mmsc.com.my