ISSN 1009 - 3095 Journal of Zhejiang University(SCIENCE) V.1, No.3, P.300 - 305, July - Sep., 2000
http://www.chinainfo.gov.cn/periodical; http://www.zju.edu.cn;
http://lib.zju.edu.cn/English; http://www.zjupress.com

PARALLEL IMPLEMENTATIONS OF NUMERICAL SIMULATION OF WIND FLOW AROUND BUILDINGS*

CHEN Shui-fu (陈水福), SUN Bing-nan (孙炳楠)
(*Dept. of Civil Engineering, Zhejiang University, Hangzhou 310027, China*)
Received May 6, 1999; revision accepted Dec. 15, 1999

Abstract: In this work, two parallel implementation strategies for the numerical simulation of wind flow around buildings were developed based on a four-processor transputer system. The first parallel strategy is based on the functional decomposition of the problem. It is easily implementable, but the degree of parallelism is low. In the second strategy, both the functional and the domain parallelisms of the problem were utilized, so the degree of parallelism is highly increased. Numerical examples indicate that the parallel speed-ups and efficiencies achievable by the second strategy are significantly greater than those of the first one.

Key words: parallel computing; parallelization strategy; wind flow around buildings; multi-processor computer; pressure correction algorithm
Document code: A **CLC number**: V211.3, TP301.6

INTRODUCTION

With the development of computer technology and numerical methods, numerical simulation techniques have become new effective approaches for the prediction of wind velocities around buildings and wind-induced pressures on the building envelope (Murakami et al., 1988; Baskaran et al., 1993).

To carry out wind-flow simulations on multi-processor computers, the parallel versions of the original sequential algorithms, namely the parallelization strategies, have to be developed. To date, efforts to achieve parallel implementations have often concentrated on the use of domain decomposition techniques, which take advantage of the domain parallelism of the problems (Braaten et al., 1991; Schreck et al., 1993). Domain decomposition strategies are generally suitable for different problems. However, due to the necessity of domain subdivision, data communication, computer programming and other aspects of these strategies often appear complicated. In numerical simulation of wind flow around buildings, many non-linear discretized equation systems must be solved (Patankar, 1980). In practice, some of these equations may be solved concurrently instead of one by one, provided that the iterative principle of the original algorithm is not changed. This is so called the functional or task parallelism of the problem. Functional decomposition strategies, which mainly utilize the functional parallelism of the problem, do not require domain subdivision, and are easily programmed and easily achieved in comparison with domain decomposition approaches. However the implementation of these strategies largely depends on the number of different tasks which can be performed independently (Lewis et al., 1993). The degree of parallelism of the strategy may be too low in some cases.

In this work, two parallelization strategies, namely the functional decomposition (FD) strategy and the dual decomposition (DD) strategy, were developed for the simulation of two-dimensional wind flow around buildings. The parallelization was based on a four-processor transputer system. The FD strategy is easily implementable, but the degree of parallelism is relatively low. In the DD strategy, both functional decomposition and domain decomposition techniques are adopted, and the degree of parallelism has been highly raised. The achieved parallel efficiencies of the strategy are also significantly increased.

* Project (59478034) supported by Natural Science Foundation of China.

TRADITIONAL SOLUTION ALGORITHM

Consider a steady two- or three-dimensional wind flow around buildings. If the k-ε turbulence model is used, the conservation differential equations governing the transport of mass, momentum in each direction, turbulent kinetic energy k, and its dissipation ε, can be written as the following general form (Patankar, 1980)

$$\frac{\partial}{\partial x_j}(\phi U_j) = \frac{\partial}{\partial x_j}\left(\Gamma_\phi \frac{\partial \phi}{\partial x_j}\right) + S_\phi \qquad (1)$$

where ϕ represents the velocity components U_j, k and ε. Γ_ϕ is the effective diffusion coefficient and S_ϕ the source term for the variable ϕ. Using the staggered grid system and the control volume method (Patankar, 1980), Eq. (1) is then discretized into the following form

$$a_P\phi_P = \sum_{nb} a_{nb}\phi_{nb} + b_P \qquad (2)$$

where subscript P is the control volume center on which the dependent variable is computed, and nb refers to the nearest neighboring control volume centers.

For a two-dimensional wind flow problem, Eq. (2) is a series of coupled nonlinear algebraic equation sets with five dependent variables U, V, P, k and ε. On single-processor computers these equation sets may be solved sequentially using the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) pressure-correction algorithm (Patankar, 1980). The discretized momentum equations for U and V are solved firstly using the latest or initially assumed values of all variables. The calculated velocity components U and V generally do not satisfy the continuity equation. They need to be corrected by adding a pressure gradient correction. Inserting these velocity corrections into the discretized continuity equation leads to a pressure correction equation, from which pressure correction P' can thus be solved. The velocities and pressures are corrected using the obtained pressure correction and the transport equations for k and ε are then solved. This completes one outer iteration of the SIMPLE algorithm. The above procedure is repeated until convergence is achieved.

PARALLEL IMPLEMENTATIONS ON MULTI-PROCESSOR COMPUTERS

The parallel computer used in this study is a distributed memory MIMD system with four IMS T800 transputers processors. Each transputer has four physical links which can be connected to the other processors. One of the transputers is connected to the host, which is called Root processor.

1. Functional decomposition (FD) strategy

For a two-dimensional problem of turbulent wind flow around buildings, five discretized transport equations for U, V, P', k and ε need to be solved at each outer iteration of the SIMPLE procedure. According to the functional parallelism of the algorithm, it is proposed that the U, V, k *and* ε equations are first assigned to the four processors respectively, and solved independently on separate processors. Then the values of U, V, k and ε are updated. These updated values are used for the calculation of the coefficients of the pressure correction P' equation. The P' equation is then solved sequentially on one processor and the velocity components and pressures are corrected using the calculated pressure correction. This completes one outer iteration of the FD parallel strategy.

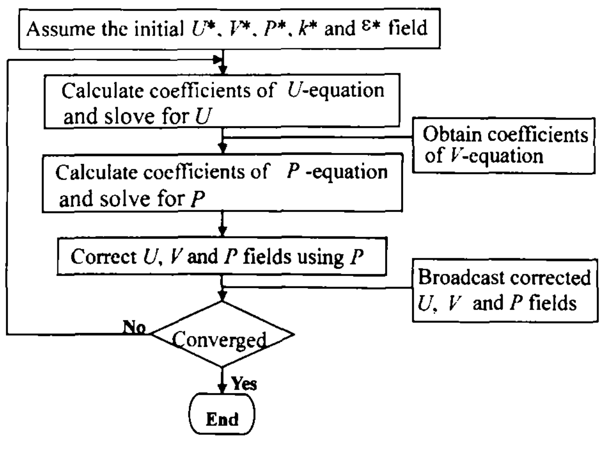

Fig. 1 Solution and communication procedure of the FD strategy (Root processor)

The FD strategy does not require domain subdivision (i. e. all outer iterations are still carried out in the whole solution domain), so it is easily implementable. The strategy has fewer solution steps but contains coarse-grain tasks, data communication of the strategy is also simple. Fig. 1 shows the solution and communication procedure of the Root processor. Although the coefficients for the U, V, k and ε equations are calculated using the previous iteration values, the updated velocity-equation coefficients are used in the calculation of the P' equation coefficients. Therefore, the iterative principle of the original SIMPLE algorithm is largely maintained. In this strategy, however, as P' equation is solved on one processor, during which the other three processors are idle, the degree of parallelism is relatively low. This will lead to a low maximum theoretical speed-up and a considerable loss of parallel efficiency. A modified parallelization strategy based upon both functional and domain parallelism will be proposed in the next subsection.

2. Dual decomposition (DD) strategy

In this strategy, the U, V, k and ε equations are first solved concurrently and independently on four processors. The solution domain is then divided into four subdomains, and each subdomain is assigned to one processor. The P' equation is solved concurrently on four processors and the velocities and pressures are corrected in parallel using the computed pressure correction. This completes one outer iteration of the DD strategy.

Fig. 2 shows the solution and communication procedure of the Root processor using the DD strategy. In this strategy, the solution of P' equation and the updating of velocity and pressure values are performed concurrently on four processors. The degree of parallelism, therefore, is considerably increased in comparison with that of the FD strategy, and the parallel efficiency achievable can be significantly raised.

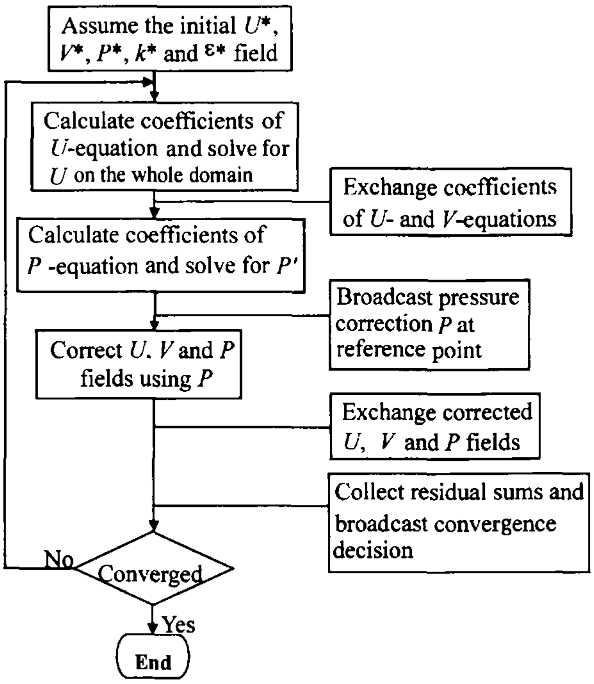

Fig. 2 Solution and communication procedure of the DD strategy (Root processor)

PARALLEL EFFICIENCY ANALYSIS

Parallel efficiency is considered as an important parameter for measuring the performance of different parallelization strategies. It is defined as

$$E_p = \frac{S_p}{n} = \frac{T_s}{nT_p} \tag{3}$$

where S_p is the corresponding speed-up, T_s is the total execution time on a single processor using the sequential algorithm and T_p is the total execution time on n processors using the parallelization strategies.

In numerical simulation of wind flow around buildings, the loss of parallel efficiency is mainly due to the following factors:

(1) the sequential run-time portion of the code,

(2) idle time for processors caused by load imbalancing,

(3) time cost for local and global communication,

(4) increase in the number of iterations caused by the degradation in convergence rate over the fully sequential algorithm.

The effect of factor (1) generally results in a maximum theoretical speed-up within which the speed-up attainable is limited. The maximum theoretical speed-up can be determined by Amdahl's law (Amdahl, 1967), which is defined as

$$S_{p\max} = \frac{n}{F_s \times n + F_p} \tag{4}$$

where F_s and F_p are the sequential and parallel portions of the code, respectively. The maximum theoretical speed-ups and efficiencies attainable for the two proposed strategies are listed in Table 1. It is seen that the maximum theoretical speed-up of the DD strategy is significantly increased in comparison with that of the FD strategy. This is due to the minor portion of the sequential run-time and high degree of parallelism of the DD strategy.

Table 1 Maximum theoretical speed-ups and efficiencies for different strategies

Parallelization strategy	Number of processors	Maximum theoretical speed-up	Maximum theoretical efficiency
FD strategy	4	2.20	55.00%
DD strategy	4	3.54	88.50%
Sequential algorithm	1	1.00	100.00%

At each outer iteration of the FD and DD strategies, owing to the rough equality of the solution throughputs of the U, V, k and ε equations, the calculations on the four processors are approximately synchronous. During the solution of the P' equation and the correction of velocities and pressures in the DD strategy, the processors can also be easily synchronized by ensuring that the four subdomains have the same number of control volumes. Therefore the effect of factor (2) on the loss of efficiency for the two strategies is quite low. In the DD strategy, the P' equation is solved on four processors, during which some more data exchange is needed. The loss of efficiency due to communication, therefore, will be slightly higher than that in the FD strategy. Although the two parallelization strategies have largely maintained the iterative principle of the original algorithm, the functional and domain decomposition will lead to a degradation in the rate of convergence and hence result in an increase in the number iterations.

NUMERICAL EXAMPLES

Numerical simulations of wind flow around the Albertslund house (Selvam, 1990) were carried out on a four-processor transputer system. In the simulation, the house was assumed to be a two-dimensional rectangular building model with a length of 1.50m and a height of 1.63m. The computational domain was also rectangular and was discretized into a structured grid system. The computed mean streamlines around the building are shown in Fig. 3. The distribution of mean velocities around the house has been roughly well predicted as seen in this figure.

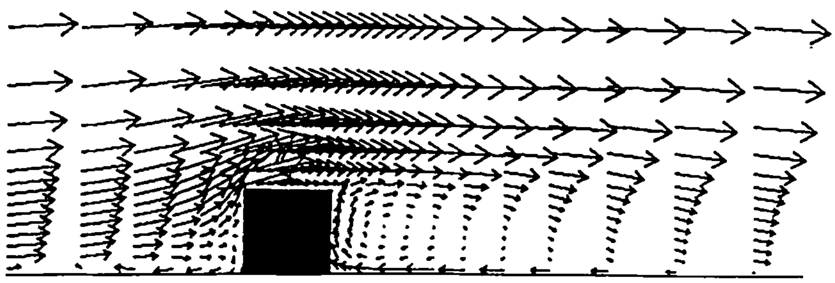

Fig.3 Computed mean streamlines around the house

Table 2 shows the numbers of outer iterations and the total run-times of the two parallel strategies together with the sequential algorithm. It is seen that the number of outer iterations required for convergence is increased by about 2.05% in the FD strategy and 3.98% or so in the DD strategy compared with that in the sequential SIMPLE algorithm. This indicates that use of the two parallelization strategies resulted in slight degradations in convergence rate.

Table 2 Outer iteration numbers and total run-times for different strategies (54 × 30 **grid**)

Parallelization strategy	No. of processors	No. of outer iterations	Total computing time (s)
FD strategy	4	795	4722.76
DD strategy	4	810	3318.72
Sequential algorithm	1	779	8530.10

Table 3 Speed-ups and parallel efficiencies for different strategies (54 × 30 **grid**)

Parallelization strategy	No. of processors	Speed-up	Parallel efficiency(%)
FD strategy	4	1.81	45.16
DD strategy	4	2.57	64.26
Sequential algorithm	1	1.00	100.00

The parallel speed-ups and efficiencies achieved for different strategies are presented in Table 3. The speed-up of the DD strategy reached to about 72.6% of its theoretical maximum speed-up (see Table 1), and for the FD strategy, it was about 82.1% of its theoretical maximum one. Although slightly more outer iterations and communication are required for the DD strategy compared to those for the FD strategy, owing to the lesser portion of sequential runtime and the far higher maximum theoretical speed-up, the DD strategy yielded much better speed-ups and efficiencies than the FD strategy.

Fig.4 shows the effect of grid size on the parallel efficiency of the FD and DD strategies. For the FD strategy, the efficiency remains roughly the same with increased grid size. For the DD strategy, however, the efficiency slightly increases with increased grid size. For example, the efficiency on a 72 × 40 grid size is about 66.7%, which is increased about 3.75% over that on the 54 × 30 grid size. This effect is mainly due to the local communication between the neighboring subdomains, in which only the boundary values are exchanged. The number of boundary grids is increased one order less slowly than that of the total grids. As a result, the ratio of communication time to calculation time is reduced as the grid is refined in two directions and the efficiency is increased. For the FD strategy, however, this kind of local data exchange does not exist, so the efficiency remains almost the same.

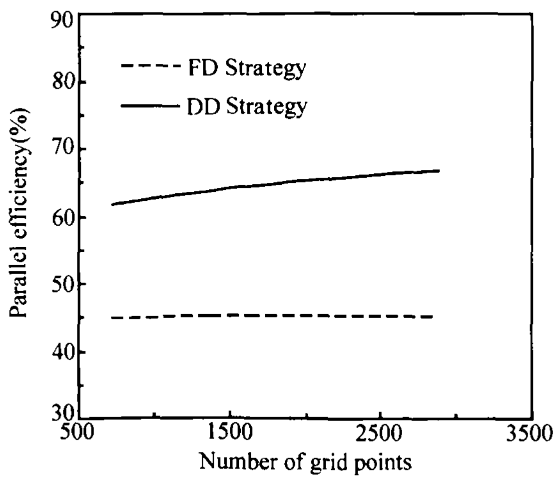

Fig.4 Effect of grid size on parallel efficiency

CONCLUDING REMARKS

There exists inherent functional and domain parallelism in the problem of numerical simulation of wind flow around buildings. A functional decomposition parallel strategy based on a four-processor transputer system is first presented in this paper. This strategy is easily implemented, but contains more portion of sequential run-time and the maximum theoretical speed-up is relatively low. A dual decomposition strategy is then proposed, in which both the functional and the domain decomposition techniques are used. The degree of parallelism of this strategy is thus greatly increased and the maximum theoretical speed-up and efficiency are comparatively high. Numerical examples indicate that the speed-ups and efficiencies achieved are significantly greater than those of the functional decomposition strategy. Furthermore, when the grid system of the computational domain is refined, the efficiency of the dual decomposition strategy is slightly increased, but for the functional decomposition strategy, it remains roughly the same.

References

Amdahl, G. M., 1967. Validity of the single processor approach to achieving large scale computer capabilities. Proc. AFIPS Spring Joint Computer Conf., 30, Atlantic City, NJ.

Baskaran, A. and Stathopoulos, T., 1993. Numerical computation of wind pressures on buildings. *Computers & Structures*, **46**(6): 1029 - 1039.

Braaten, M. E., 1991. Development of parallel computational fluid dynamics algorithm on a hypercube computer. *Int. J. Numer. Methods Fluids*, **12**: 947 - 963.

Lewis, A. J. and Brent, A. D., 1993. A comparison of coarse and fine grain parallelization strategies for the SIMPLE pressure algorithm. *Int. J. Numer. Methods Fluids*, **16**: 891 - 914.

Murakami, S. and Mochida, A., 1988. 3-D numerical simulation of airflow around a cubic model by means of the k-ϵ model. *J. Wind Engng. Indus. Aerodyn.*, **31**(2): 283 - 303.

Patankar, S. V., 1980. Numerical Heat Transfer and Fluid Flow, Hemisphere, New York.

Schreck, E. and Peric, M., 1993. Computation of fluid flow with a parallel multigrid solver. *Int. J. Numer. Methods Fluids*, **16**: 303 - 327.

Selvam, R. P., 1990. Computer simulation of wind load on a house. *J. Wind Engng. Indus. Aerodyn.*, **36**: 1037 - 1046.