US - KOREA - JAPAN Trilateral Seminar
on Frontier R&D for Constructed Facilities
21 – 23, Oct. 1991, Honolulu, Hawaii

THE FUTURE OF CFD IN CIVIL ENGINEERING: LARGE-SCALE COMPUTATION WITH VECTOR AND MASSIVE PARALLEL COMPUTERS

S. Murakami *

Summary

The concept of turbulence modelling is discussed in relation to the turbulence scale and the fineness of grid discretization of the flowfield. Then it is clarified why computational fluid dynamics (CFD) requires a large amount of CPU time. Next, the current status and future potential for numerical simulation of turbulent flow in civil engineering is described. Lastly it is concluded that the massive parallel computer is most promising for achieving the much higher computing speeds requisite for future stages of CFD. Furthermore the simulation results of room airflow by parallel computing is shown.

1. Introduction —— development of CFD supported by evolution of supercomputer

In the field of fluid dynamics, new analytic methods have been developed rapidly in recent years as a result of the burgeoning growth of computer hardware technology, in particular the development of the supercomputer using a vector processor. This paper addresses some aspects of the current status and future prospects of computational fluid dynamics in the field of civil engineering, using examples mainly from wind engineering.

Computational methods originated in the field of aeronautics and now have a history of more than 30 years. Numerical methods for simulating turbulent flows have been developed mainly in the fields of mechanical and aeronautical engineering during the past several decades by Spalding, Launder and others. However, airflow in wind engineering problems is far more complicated. In many cases, the current CFD technology borrowed from other fields is thus inadequate and new research and development are urgently required in this field.

CFD is a large system composed of a great many subsystems ranging from computational mathematics to the fluid dynamics of turbulence statistics. In this paper, the large amount of computation required in CFD is discussed from the viewpoints of grid discretization, turbulence models and types of computer hardware.

In CFD as elsewhere, all numerical analyses are conducted using discretized values of physical properties. The accuracy of the numerical predictions thus depends on the fineness of the grid discretization. Therefore CFD developed by increasing the number of grid points, a development supported by advances in computer performance. Fig.1 illustrates the progress in computing speed, which is now at the performance level of 20 Giga FLOPS. This progress has been achieved mainly by the development of a supercomputer using a vector processor. However, the vector-type supercomputer has its limitations and further progress in computing speed will be made by the massive parallel computer, as will be discussed later in detail. Fig.2 shows the increase of grid points used for various large-scale computations in each stage of CFD development. The greatest number of grid points ever used was about 7 million.

In the following sections, the reasons why CFD requires such a large number of grid points will be explained and methods for overcoming this difficulty will be discussed in relation to turbulence modelling.

---

* Professor, Institute of Industrial Science, University of Tokyo, JAPAN

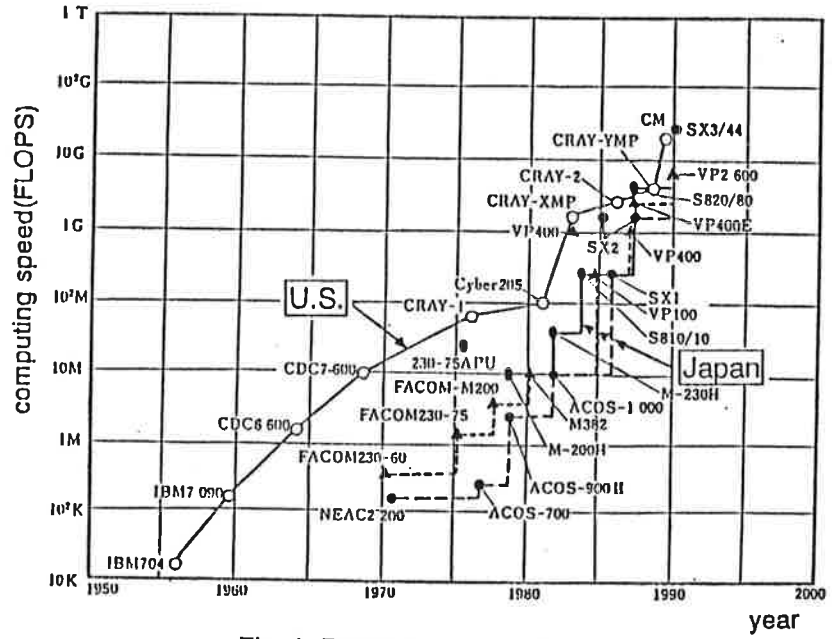


Fig. 1 Progress in computing speed

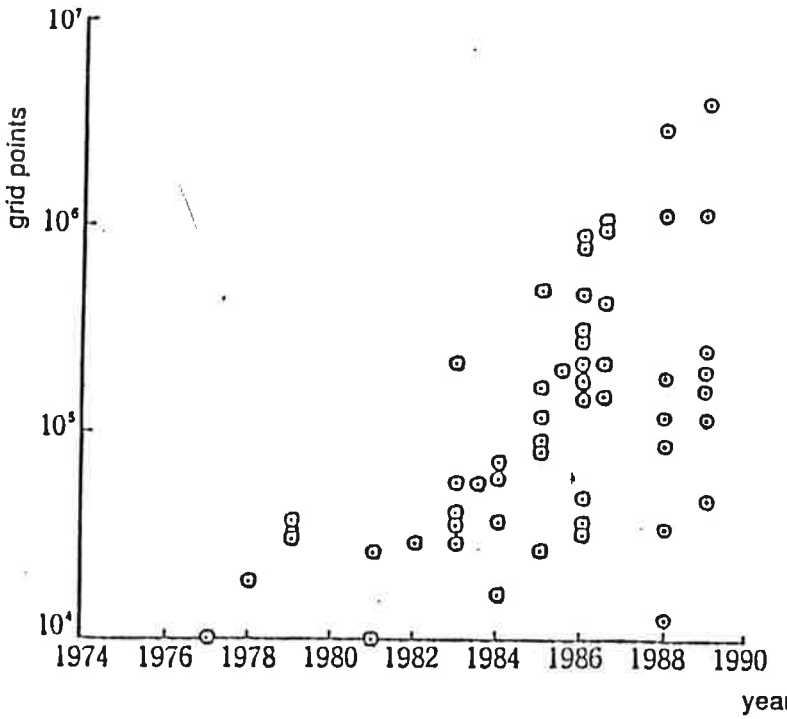


Fig. 2 Increase in number of grid points used in large-scale computation in CFD

2. Kolmogorov microscale and grid discretization

The flowfields treated in civil engineering are usually expressed by the incompressible Navier–Stokes equations (hereafter N – S equations), which are shown in the Appendix. This set of nonlinear partial differential equations is infamous for the difficulty with which they are solved. Furthermore, the flows treated in this field, for example atmospheric wind flow or airflow in an air-conditioned room, are always highly turbulent, and we must thus rely on the theory of turbulence statistics when solving these N – S equations by means of numerical methods.

Turbulent flowfields are characterized by various scales of turbulent eddies. Hence the most important decision in numerical simulation is how finely we should resolve these eddies in numerical analysis. The amount of CPU time is completely dependent on this judgement. Resolving small eddies requires much CPU time. The smallest eddy in turbulent flow, the so-called "Kolmogorov microscale η" is very small, around 0.1～1mm for natural wind or room airflow. It will be evident that discretization of the flowfield corresponding to the Kolmogorov microscale η, is very difficult in practice.

Numerical analysis of flowfield, whether the numerical method be finite difference or finite element, begins with the discretization of the flowfield ; computation is conducted using these discretized expressions in the N－S equations etc. While the amount of CPU time increases with greater numbers of these discretizations of space and time, the accuracy of the prediction is governed by the fineness of discretization. In Fig.3 two numerical results are compared to illustrate the effect of coarse and fine grids. When grid discretization becomes finer, as in case 2, solution error decreases dramatically (Caruso et al.(1986), Murakami et al.(1990a)). The effect of grid discretization on prediction accuracy is so significant that grid design is one of the most important aspects of CFD.

As was stated above, turbulent flow can be reproduced with complete accuracy only if the flowfield is discretized on the scale of Kolmogorov microscale η. However, such a fine discretization is almost impossible to obtain in practice. The size of the microscale (η/l, l : integral scale) decreases as the Reynolds number (ul/ν, hereafter Re) increases. Consequently the number of grid points must be increased in proportion to the value of $Re^{9/4}$, according to the theory of turbulence statistics. This means that a huge number of grid points would be needed for the simulation of a flowfield with a certain magnitude of Re. The number of grid points required for 3－dimensional analysis can be estimated as follows.

Reynolds number	10^4	10^6	10^8
Number of grid points (3 – dimensional)	10^8	10^{14}	10^{18}

note : $Re=\frac{ul}{\nu}$ u : characteristic velocity scale for turbulence
l : characteristic length scale for turbulence(integral scale)
ν : kinematic viscosity

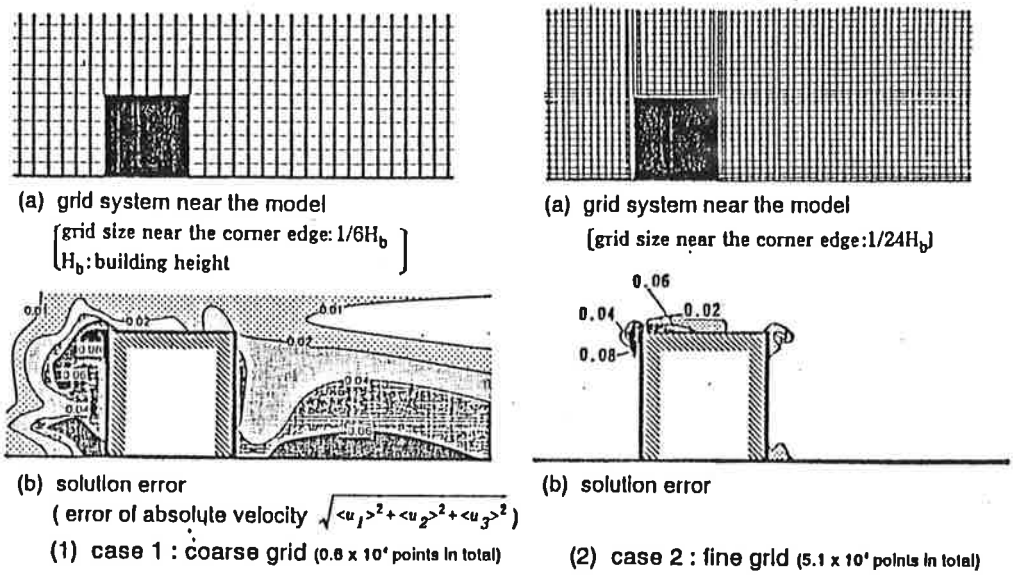


Fig. 3 Relation between grid discretization and solution error

Table 1 Ranges of grid points and CPU time used in engineering applications

	grid points	CPU time (hours)*
Flow around bluff-body	(x) (y) (z) k- ε $50 \times 49 \times 28 \fallingdotseq 6.8 \times 10^4$ ASM $50 \times 49 \times 28 \fallingdotseq 6.8 \times 10^4$ LES $63 \times 49 \times 34 \fallingdotseq 1.0 \times 10^5$	0.5 ~ 5 5 ~ 20 20 ~ 40
Flow around automobile	k- ε $100 \times 45 \times 35 \fallingdotseq 1.6 \times 10^5$	0.5 ~ 2
Room airflow	k- ε $21 \times 32 \times 15 \fallingdotseq 1.0 \times 10^4$ ASM $21 \times 32 \times 15 \fallingdotseq 1.0 \times 10^4$ LES $42 \times 54 \times 50 \fallingdotseq 1.0 \times 10^5$	0.5 ~ 3 5 ~ 10 20 ~ 40
Jet	LES $104 \times 84 \times 24 \fallingdotseq 2.1 \times 10^5$	10 ~ 30
Flow with diffuser (2D)	k- ε $151 \times 51 \fallingdotseq 0.7 \times 10^4$	0.3 ~ 1.0

*These simulations were conducted at I.I.S., University of Tokyo. CPU time presented here was measured on FACOM VP-100 (peak performance of VP-100 is about 285 MFLOPS).

The numerical method based on this concept of microscale discretization is called "Direct Numerical Simulation" (hereafter DNS, Kim et al.(1987)). While this simple and clear method uses no turbulence modelling, it requires a huge number of grid points if the Reynolds number of the flowfield is large. The Reynolds number of the flowfields treated in civil engineering is about $10^5 \sim 10^7$, so DNS of such a flowfield requires around $10^{11} \sim 10^{16}$ grid points. On the other hand, the maximum number of grid points ever used is around $10^6 \sim 10^7$, as is shown in Fig.2. Therefore it can easily be understood why it is practically impossible to solve a fully turbulent flow by means of DNS even if computer performance is advanced greatly in the future. The largest Reynolds number (ul/ν) of a flowfield ever analyzed by DNS is about $10^2 \sim 10^3$. The range of grid points used in engineering CFD is shown in Table 1.

3. Turbulence models – current status and future potential

The numerical analysis of turbulent flow has a serious obstacle of the Kolmogorov microscale, which concept was given from the theory of turbulence statistics. In order to overcome this obstacle, various types of turbulence models have been invented. Turbulence modelling is designed to simulate the averaged flowfield (coarse graining, cf.Fig.5) rather than the original flowfield. Such small scale eddies, which are difficult to resolve, are thus neglected in the process of coarse graining.

There are two types of coarse graining ; time averaging and space averaging. These types of averaging and turbulence models are illustrated in Fig.4, including DNS, which uses no turbulence model. By averaging a flowfield with steep fluctuations, the averaged fluctuations of the flow variables become less – steep and the relatively coarse grid discretization can resolve the flowfield with certain accuracy. The small eddies neglected by coarse graining can be incorporated in the simulation through the turbulence modelling.

1) Turbulence models based on Reynolds-averaged equations

The ensemble (or time) averaged forms of N – S equations are called the Reynolds equations. These equations express only the movements of large scale eddies. Since the flowfield contains only larger eddies, numerical analysis can be conducted with a rather coarse grid discretization. Most CFD simulations in the field of engineering have been based on this type of modelling partly because of inadequate computer performance. The most famous and most widely-used is "the k-ε two equation turbulence model" (hereafter k-ε model or k-ε, Launder and Spalding (1972), Murakami et al.(1988a)). However, it has become clear that the k-ε model possesses some shortcomings. The

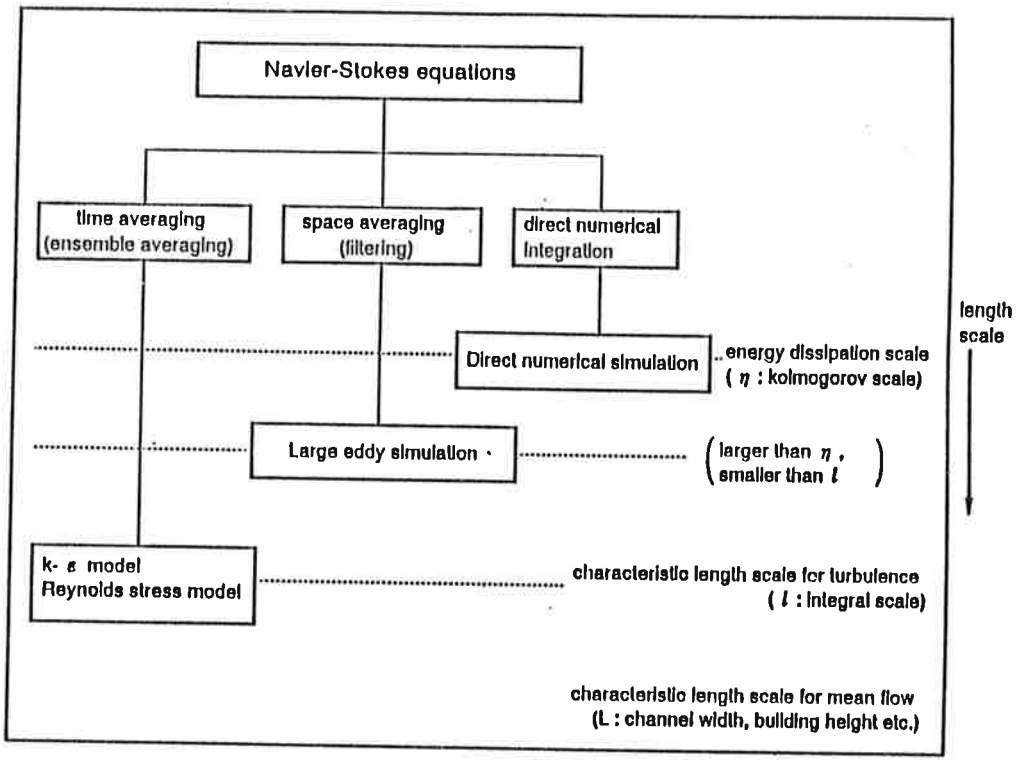


Fig. 4 Methods of numerical simulation and turbulence models

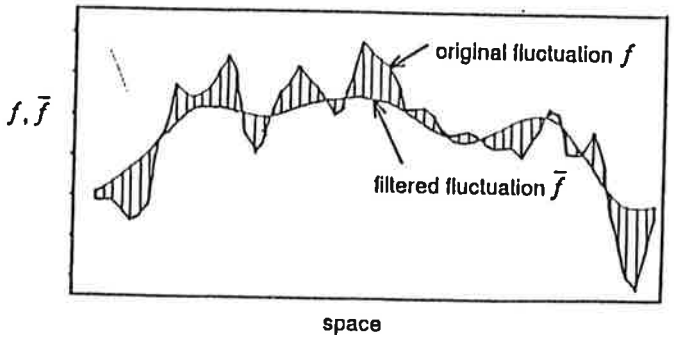


Fig. 5 Filtering of flowfield

high-precision version of the Reynolds-averaging approach is the Reynolds Stress Model (hereafter RSM). Applications of RSM and its simplified version, ASM (Algebraic Stress Model, Launder et al. (1975), Rodi (1976), Murakami et al.(1990b)) are in the tendency of increase year by year. The basic equations for these models are shown in Appendix 1 and 2.

The k-ε model has made significant achievements in the simulation and prediction of rather simple flowfields, such as channel or pipe flows. However it is clear that the standard k-ε model fails to predict flowfields with highly anisotropic properties, because the k-ε model is based on the isotropic eddy viscosity model. The flowfields treated in wind engineering are usually highly anisotropic and in flowfields around bluff bodies include separation, recirculation etc. (cf.Fig.6). Since the k-ε model has serious limitations when applied to such flowfields, it is recommended that RSM, ASM or Large eddy simulation (mentioned later in detail) be used in order to predict flowfields with higher accuracy.

The results of predictions given by various turbulence models are compared later in Figs.9 and 10. The results given by the k-ε model are clearly inferior to those of LES.

2) Turbulence models based on space-filtering

Another method for coarse graining is "space-filtering", the concept of which is illustrated in Fig.5. The degree of averaging can be adjusted easily by changing the width of the filter.

Since eddies smaller than the width of the filter (eddies of subgrid scale) are not analyzed directly and eddies larger than the filter width (eddies of resolvable scale) are simulated, this type of modelling is called "Large Eddy Simulation" (hereafter LES, Smagorinsky (1963), Deardorff (1970), Schumann (1975), Moin et al.(1982), Murakami et al.(1987)). The basic equations for LES are shown in Appendix 3. In LES, the effect of small eddies neglected by filtering is incorporated in the simulation through the turbulence model of subgrid scale eddy viscosity. Since Reynolds-averaging is not applied to the basic equations used in LES, the scale of resolved eddies in LES is usually much smaller than those of the k-ε model or RSM. Thus LES predicts a time-dependent flowfield with small fluctuations. Consequently the effect of modelling in subgrid scale on the turbulence structure of the flowfield is relatively small, compared to that of k-ε. In fact, LES has succeeded in reproducing the properties of an highly anisotropic flowfield, as shown in Figs.9 and 10.

In recent years, high-precision versions of subgrid scale modelling for LES have been developed in the field of meteorology, for example, the subgrid scale one-equation model or the subgrid scale Reynolds stress model.

To sum up these various surveys, LES is most promising for the simulation of flowfields treated in wind engineering. The weak point of LES is the large amount of CPU time required for computation, which is much larger than that for k-ε. While the k-ε model usually gives a mean flowfield LES must reproduce and store a long sequence of time-dependent flowfields. However this weak point will be overcome to a certain degree by future growth in computer hardware technology.

4. State of the art of turbulent flow simulations in wind engineering

As was mentioned above, the highly anisotropic nature of flowfields around buildings creates many difficulties in the simulation of such flowfields. The anisotropic structure of the turbulent flowfield around a building model is expressed schematically in Fig.6.

The flowfield around a building complex as visualized by a wind tunnel experiment is shown in Fig.7 (Murakami et al.(1991a). The prediction results of complicated flowfields around a building complex are illustrated in Fig.8 (Murakami et al.(1988b)).

The distributions of velocity vector and turbulence energy predicted using various turbulence models are given in Figs.9 and 10. As shown in Fig.9, the reverse flow on the roof is not reproduced in the case of k-ε . Furthermore k-ε overestimates the value of k near the frontal corner as shown in Fig.10. The prediction given by LES is most successful among the results given by the various turbulence models. The discrepancies observed in k-ε are significantly improved in the results of ASM (Murakami et al.(1991c)).

Fig.11 shows the prediction of wall surface pressure. The results of LES agree very well with the result of the wind tunnel test. The degree of accuracy of the prediction is satisfactory from the viewpoint of engineering application.

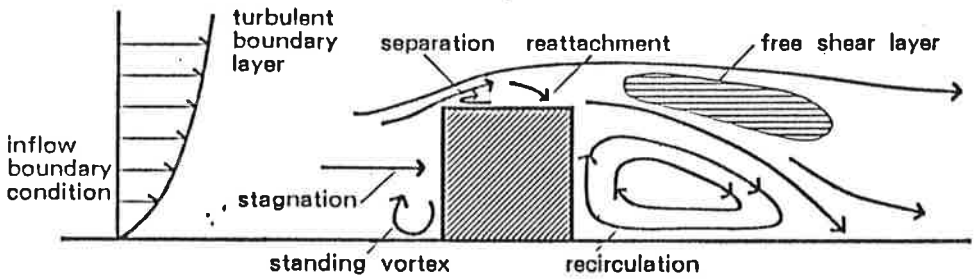


Fig. 6 Flowfield around a building model

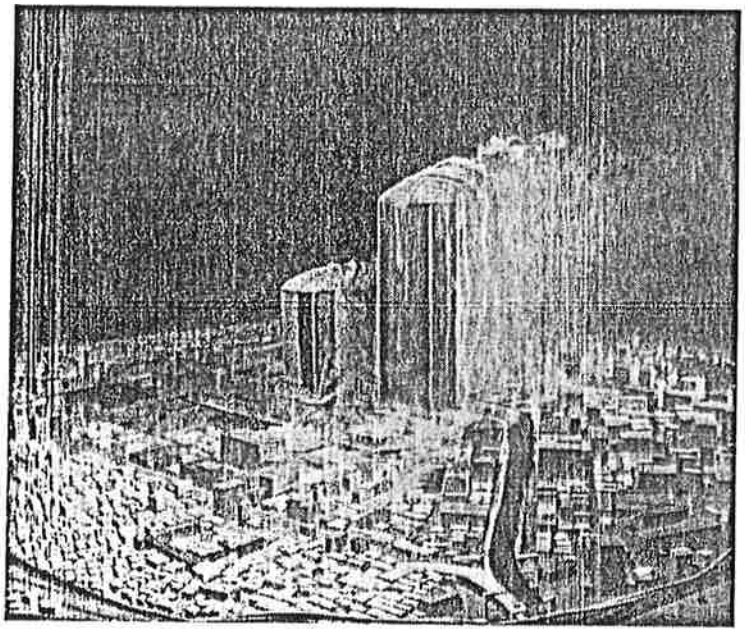
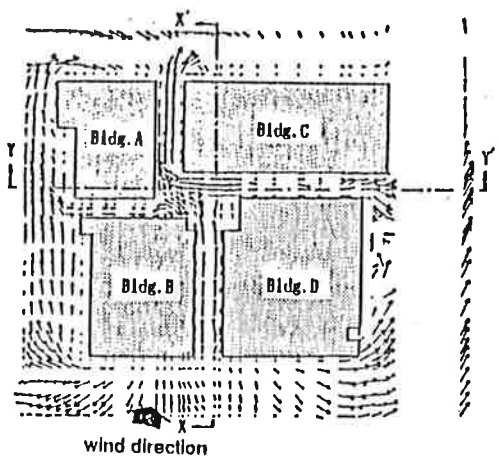
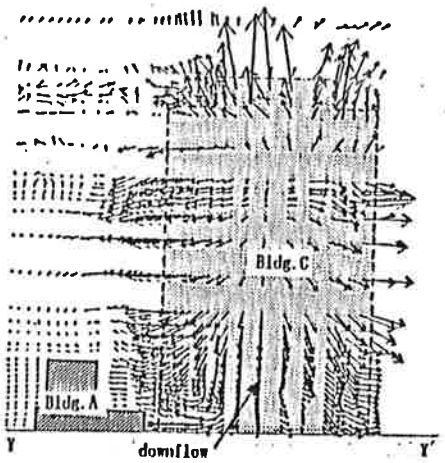


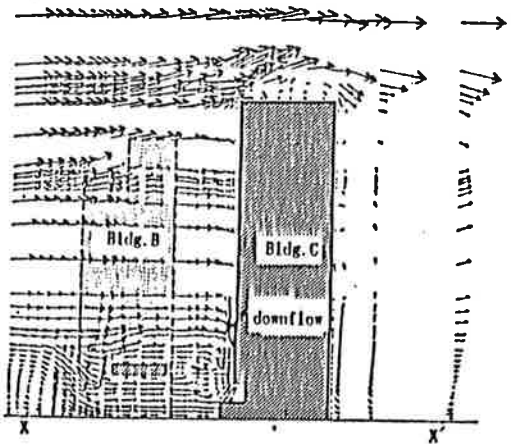
Fig. 7 Airflow around an actual building complex visualized in a wind tunnel



(1) velocity vectors at ground level (Z=1.5m)

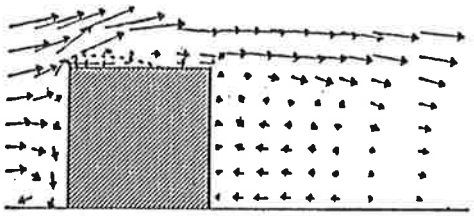


(3) velocity vectors at vertical plane (Y-Y' plane)

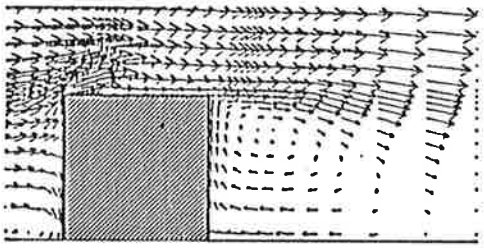


(2) velocity vectors at vertical plane (X-X' plane)

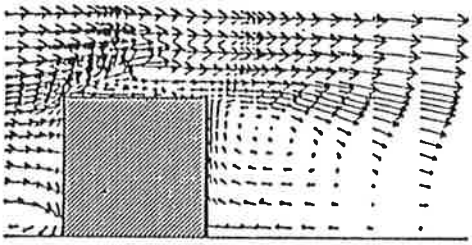
Fig. 8 Prediction of flowfields around an actually planned building complex [by k-ε model with grid points of 52(x)×55(y)×42(z)]



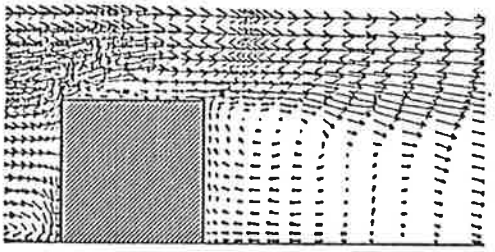
(a) wind tunnel experiment



(b) k- ε model

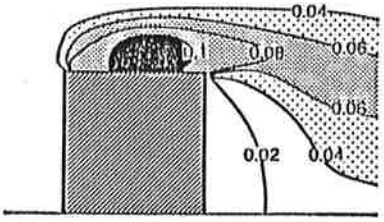


(c) Algebraic Stress Model (ASM)

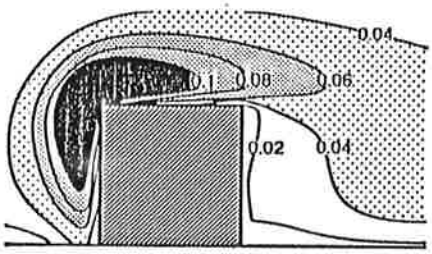


(d) Large Eddy Simulation (LES)

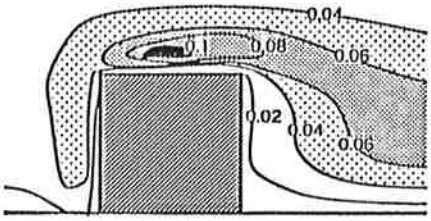
Fig. 9 Prediction of velocity vectors by various turbulence models



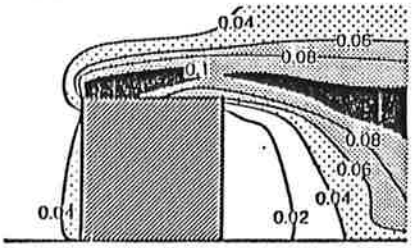
(a) wind tunnel experiment



(b) k- ε



(c) ASM



(d) LES

Fig. 10 Distributions of turbulence energy k

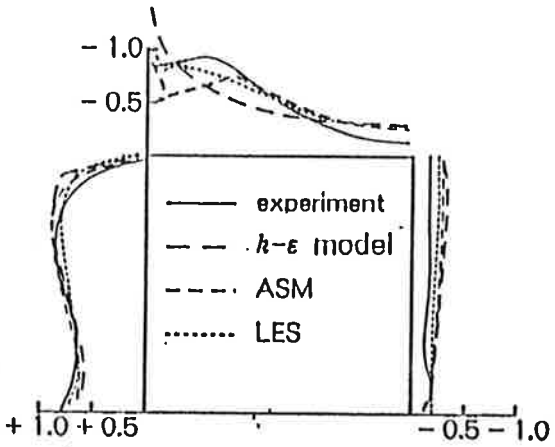


Fig. 11 Mean surface pressure coefficient $\langle C_p \rangle$

[• grid discretization (for Figs. 9, 10 and 11)
50(x)×49(y)×28(z) (k-ε and ASM)
63(x)×49(y)×34(z) (LES)]

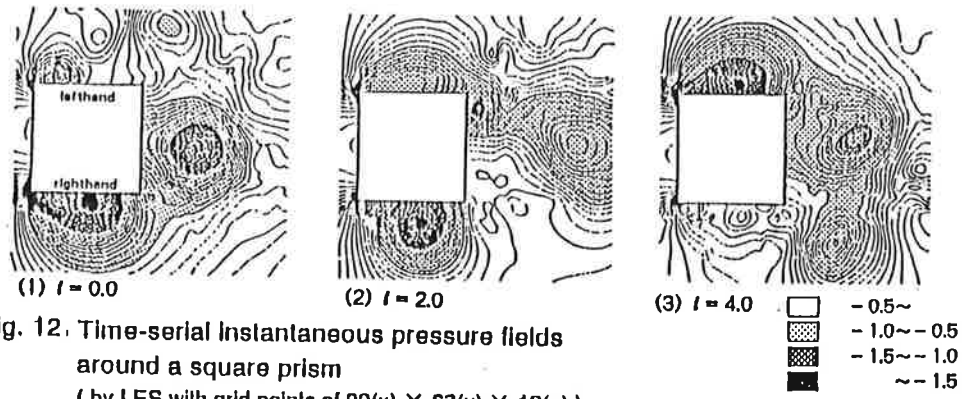


Fig. 12 Time-serial instantaneous pressure fields around a square prism
(by LES with grid points of 99(x) × 63(y) × 10(z))

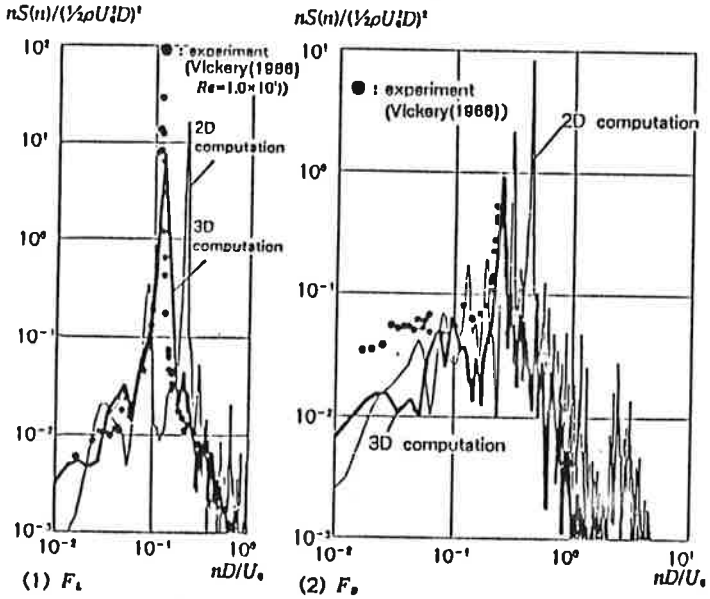


Fig. 13 Spectrum of fluctuating lift force F_L and drag force F_D by LES
[• grid discretization: 99(x)×63(y) (2D computation),
99(x)×63(y)×10(z) (3D computation)]

The instantaneous pressure field around a square prism given by LES is shown in Fig.12 (Murakami et al.(1991b). Since LES can produce the distribution of time-dependent fluctuations of the pressure field, it can make a great contribution to the development of wind engineering.

In CFD, two dimensional computation is often conducted because it requires much less CPU time in comparison with 3D computation. Furthermore, various difficulties in numerics are also greatly decreased in 2D computation. However, since 2D computation for turbulent flow often involves crude approximations, the accuracy of 2D computation must be investigated carefully.

The spectra of fluctuating lift force F_L and drag force F_D as predicted by LES in 2D and 3D computations are shown in Fig.13. The result of 3D computation corresponds very well with the experiment, but 2D computation shows some discrepancies. It may be concluded that 3D computation is recommended for accurate prediction of turbulent flowfields.

3D computation with fine grid discretization requires a large amount of CPU resources. Therefore, further development of the high performance computer is required before 3D computation becomes generally available for engineering applications. The massive parallel computer is expected to provide the high performance required for large-scale 3D computation.

5. Requirement for large-scale computing and improvement of performance supported by parallel computer

A brief history of the evolution of the computer is shown in Fig.1. The most advanced vector computer has reached the performance level of 20 Giga FLOPS (GFLOPS). The vector computer is expected to attain the level of 100 GFLOPS in the near future, computing speed regarded as the practicable limit of this evolution. The massive parallel computer is generally considered capable of breaking through this limit of computing speed. Therefore the large-scale computing of turbulent flow required in future stages will as a necessity rely on the massive parallel computer.

In science and technology there are many fields whose further development is impeded by the insufficient performance level of the supercomputer. Fig.14 shows the realized and expected applications of the supercomputer at each stage of development (Federal High Performance Computing Program, 1989). A performance level of 1 Tera FLOPS (TFLOPS) and memory of 10 to 100 Giga words will make possible many significant break-throughs using computer simulation in various techno-science fields, and turbulent flow simulation will not be an exception. A performance speed of 1 TFLOPS is a mile-stone for turbulent flow simulation by LES and DNS (with not-high Reynolds number).

Fig.15 compares the capacities of the parallel computer and the single processor computer (Ikudome,1991). A computing speed of 1 TFLOPS can be attained only by the massive parallel computer which has more than 1000 processors and distributed memories for each processor.

6. Concept of parallel computing

The concept of parallel computing did not evolve from the simple idea of enhancing computing capacity by assembling many conventional processors in parallel ; rather does it represent a fundamental modification of computer architecture, i.e. a change from Neumann-type processing to parallel-type processing (e.g. Hillis, 1985).

Conventional computers, including vector computers, are sequential processing computers, i.e. von-Neumann computers. These computers have an extremely low availability of semiconductor devices in a practical sense. Though this type of computer consists of a large number of LSI chips both in CPU and in memory, only a single chip in memory is in operation at one time because this type of computer executes only one operation at each step of the machine cycle.

Efficient utilization of these huge computer resources, i.e. the full integration of the ability of each separate semiconductor circuit, is attained only by a change in computer architecture from sequential processing to parallel processing. Even with the present technology of semiconductor devices and the present level of integration of circuits, this change will allow a remarkable evolution in computer performance. The performance of the parallel computer is actually assumed to increase in proportion to the number of processors and is expected to advance far beyond 100 GFLOPS, the limit of a vector computer. This improvement in computer performance is sure to bring about great break-throughs in high-precision CFD.

It should be noted that the development of parallel processing depends greatly on the development of software. Present hardware technology such as semiconductor devices is adequate for the needs of the parallel computer.

7. Development of software — a key for parallel processing

Procedures of analyzing physical phenomena including flow simulation can be divided into unit procedures or subtasks. These subtasks need not always be treated in a linear sequential manner and some subtasks may be processed in a non-sequential order. In this case, parallel task-processing is possible.

The linear sequential processing computer executes all subtasks in sequence, regardless of whether the subtasks can be processed in a parallel manner or not. It is thus easy to make a program

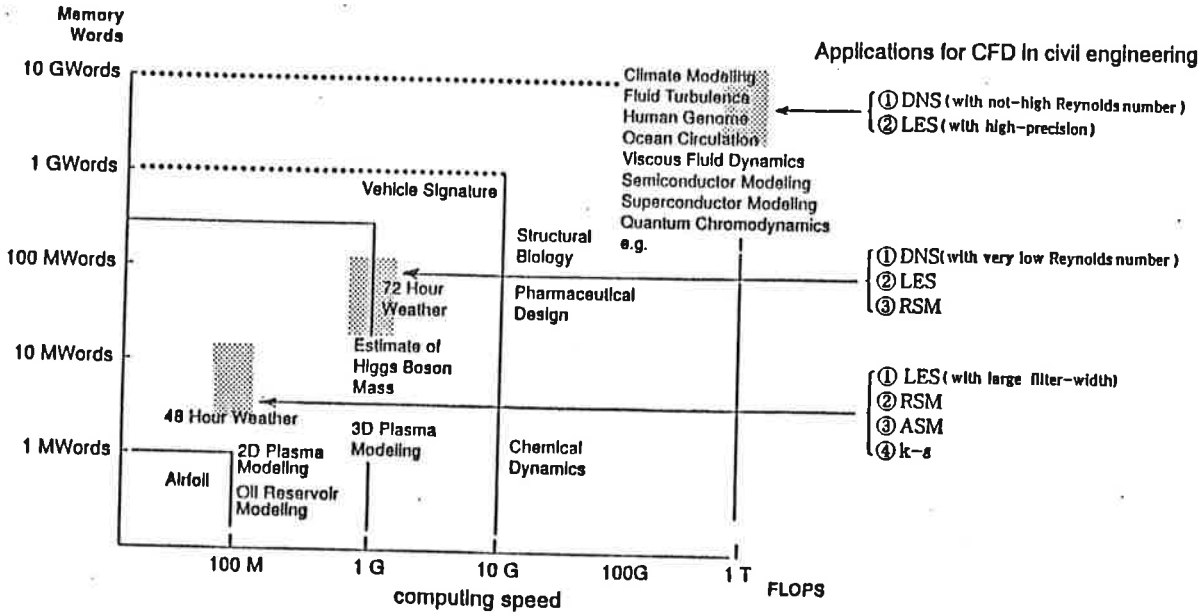


Fig. 14 Grand challenges of supercomputer, realized and expected

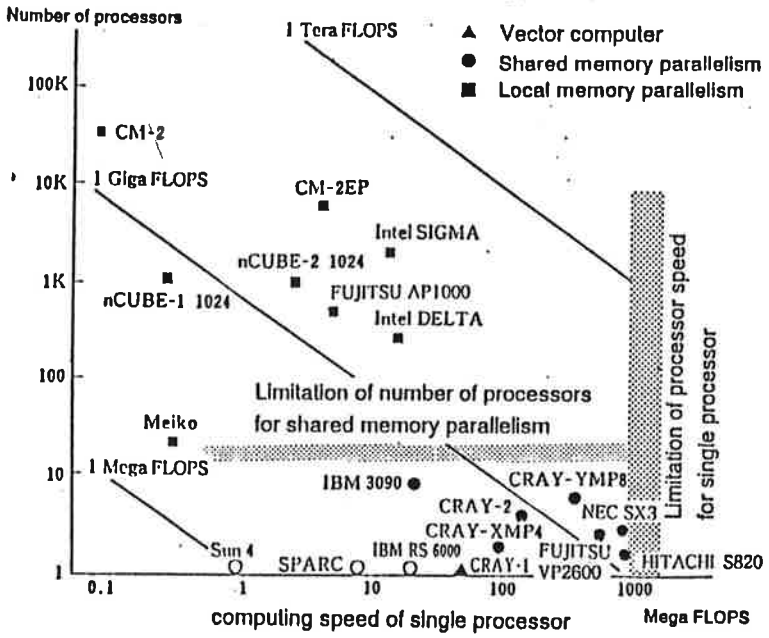


Fig.15 Progress of computing speed given by parallel computer
(Including machines under development)

code with a sequential processing computer because the programer need not concern himself with an analysis of the procedure and the possibility of parallel processing, which is rather troublesome.

While a sequential processing computer executes all subtasks in sequence, a parallel processing computer can execute all subtasks simultaneously. Synchronization is of course necessary between each parallel subtask according to the sequence of the total procedure (e.g. time marching in fluid simulation). The results of each subtask should be synthesized at each time step. The key problem to be solved in parallel processing is how to code the program for these synchronizing procedures

so as to minimize idle time for the processors.

In the coding stage the whole procedure is divided into subtasks and the system for execution is analyzed; parallel parts and their sequential relation (e.g. time marching in fluid simulation) must be designed well. Hence parallel processing requires precise analysis of the computational procedure:

(1) optimization of division (design of subtasks),

(2) synchronizations between subtasks.

This process may be likened to control of the construction schedule in civil engineering, e.g. PERT. At present a multi-purpose software which automatically analyzes the procedures, divides them into subtasks and gives the appropriate execution schedule is not available. Extensive studies must be conducted during this decade to solve this subject.

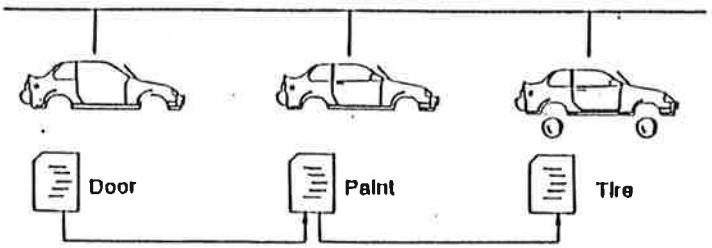
8. Task partitioning for parallel processing

There are two methods of task partitioning of parallel processing; functional partitioning and regional partitioning, as shown in Fig.16. Regional partitioning is easier and more applicable to parallel processing and synchronization of subtasks than functional partitioning (Ikudome,1991).

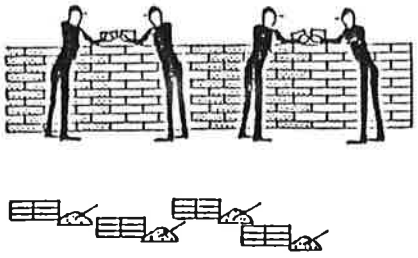
In computation, the subtasks are distributed among processors and then executed. Since a subtask cannot be completely independent of other subtasks, each processor needs to communication with the others. It should be stressed that the communication method and the time needed for communication (i.e. overhead) are very important.

Communication between processors is a very important issue to be considered when constructing a parallel processing system. Thereby various interconnection networks have been proposed, one of which is the torus network, shown in Fig.17. In the case of fluids, one fluid mass interacts densely with the surrounding masses. This property is called "the proximity effect". When regional partitioning is applied to fluid simulation, this proximity effect can save significantly on communication overhead because communication is actually done only between adjacent processors .

Subtasks should be distributed to all available processors in such a way as to make the processor loads equal and also to minimize the idling time required for synchronization. As stated above, no parallel processing system at present has the capability for automatic partitioning within the operating system nor for the dynamic partitioning which minimizes idling time during execution. At present, each user must design all of the partitioning procedures.



(1) Functional partitioning



(2) Regional partitioning

Fig.16 Task partitioning

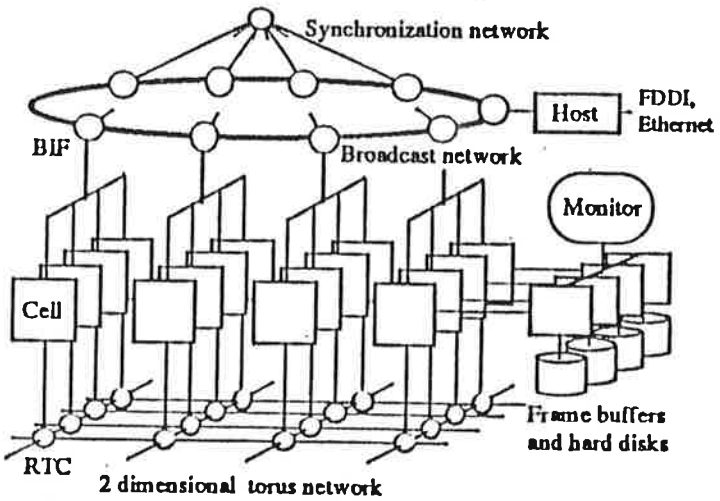


Fig. 17 Parallel computer used
(network structure of Fujitsu AP1000)

9. Advantage of CFD for parallel processing

Numerical analysis for the partial differential equations governing continuums such as fluids can be easily adapted to parallel processing with regional partitioning. In particular, for explicit solution of the time marching equation, communication between processing subtasks can be reduced significantly, and each subtask can be dealt with independently to a certain degree. A subtask only refers values adjacent to its region, according to the proximity effect, and adds only a small load to the interconnection network. Therefore it is pertinent and relatively easy to modify a program coded for sequential processing into a program for parallel processing. In this context, during the development of a general compiler language for parallel processing, the easy modification of sequential into parallel is very advantageous and can play an important role in the development of new computer hardware, which is also in transition from sequential to parallel.

10. Flow simulation by parallel computing

10.1 Parallel computer used and numerics

A schematic view of the general structure of the Fujitsu AP1000 parallel computer used for the flow simulation is shown in Fig.17. It is composed of 512 processors in total. Each cell processor has a 16MB memory, in which an operating system controlling the processor is installed. Cell processors are interconnected by a broadcast network and a torus network which control the communication and the synchronization between the cell processors. A SPARC RISC chip of 15 MIPS and 8.3 MFLOPS is used for the cell processor. The simulations here used maximum of 64 cell processors.

The configuration of the flowfield for the simulation is shown in Fig.18; two-dimensional laminar room airflow with one supply and one exhaust opening. The Reynolds number of the supply jet is 100. The governing equations are a continuity equation and N－S equations (cf.Appendix). The flow field is discretized into grid points of 242X242. The method of partitioning is regional. The partitioning in the case of 64 processors is shown in Fig.19. The ABMAC－method with Adams－Bashforth scheme is applied for time marching.

10.2 Results and discussion

Many simulations with different numbers of processors were conducted. The computing time needed for each simulation was different but the results were the same (Murakami, submitted).

The stream lines at steady state are shown in Fig.20. The ratio of computing speed to the increase in number of processors is shown in Fig.21.

The computing speed for the simulation increases as the number of processors is increased, but the computing speed increase is not strictly proportional to the increase in number of processors. Increase in the number of processors makes each task size smaller and consequently makes the communication load larger in comparison to the calculating load, hence the reduction in the speed increase ratio.

Though CFD seems to be easily adapted to parallel processing through utilization of regional partitioning based on the proximity effect, it is not so easy to conduct a fully efficient simulation through parallel computing. In this exercise, when the partitions became smaller, efficiency decreased because of the increase in communication time. However as the grid points (discretization of flowfield) became larger, this efficiency reduction became smaller. Hence the parallel computer is advantageous for large-scale computing with huge number of grid points.

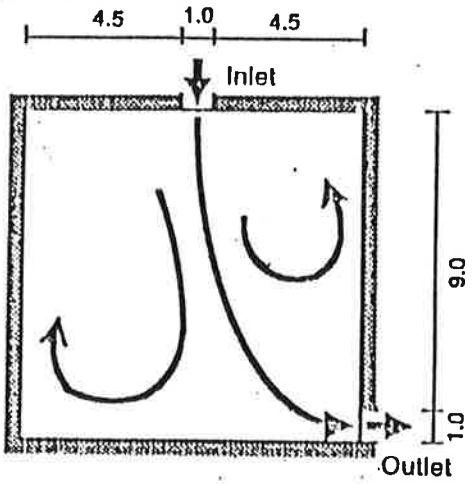


Fig.18 Flow field analyzed
(2D room airflow)

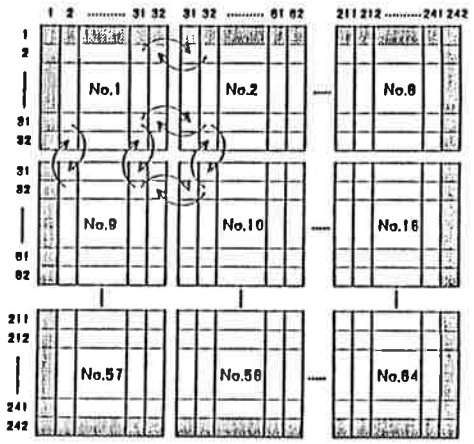


Fig.19 2-D regional partitioning of flowfield
(In case of 64 processors)

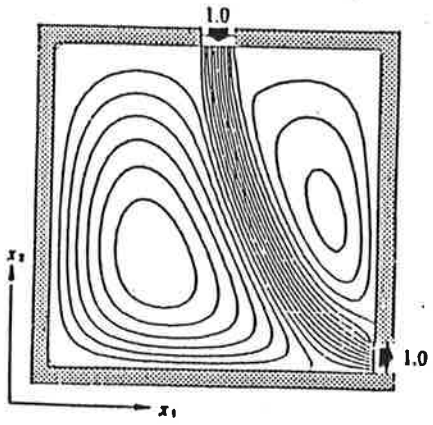


Fig.20 Result of simulation ; Stream lines

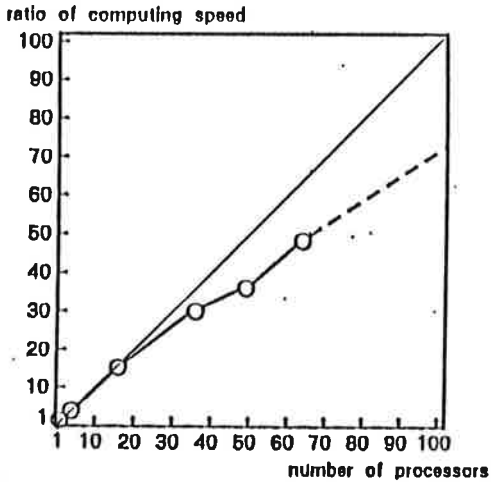


Fig.21 Ratio of computing speed
to number of processors

11. Concluding Remarks

1) Even if computer performance is greatly advanced in the future, it will remain necessary to utilize turbulence models in order to simulate the highly turbulent flowfields treated in civil engineering.
2) Large eddy simulation is most promising for the simulation of highly turbulent and anisotropic flowfields, but it needs a great amount of CPU resources for high speed and large memory.
3) Massive parallel computers are expected to attain the high performance required for high-precision CFD in the future.

Acknowledgements

The numerical study utilizing the k-ε model, ASM and LES was carried out in cooperation with Dr. A. Mochida (University of Tokyo) and Dr. Y. Hayashi (University of Tokyo). The research concerned with the massive parallel computer was conducted in cooperation with Prof. S. Kato (University of Tokyo), Prof. Y. Utsumi (Miyagi National College of Technology) and Dr. K. Mizutani (Sanken Setsubi Kogyo). All numerical exercises on the massive parallel conputer were done with Fujitsu AP1000 belonging to Fujitsu Parallel Computing Research Facilities. This paper was written with the assistance of Dr. A. Mochida, Dr. Y. Hayashi, Prof. S. Kato, Prof. Y.Utsumi, Dr. K. Mizutani and Mr. Y. Tominaga (graduate student of University of Tokyo). The author would like to express his gratitude for their valuable contributions to this work.

References

Caruso, S. C., J. H. Ferziger and J. Oliger, Adaptive grid techniques for elliptic fluid-flow problems, AIAA 24th Aerospace Science Meeting, (1986)
Deardorff, J. W., A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers, J. Fluid Mech., 41 (1970) 453-480
Federal High Performance Computing Program, U. S. A., (1989)
Hillis, W. D., The Connection Machine, MIT Press, (1985)
Ikudome, K, Super Software - Software of parallel processing -, bit (Tokyo), Vol. 23, No. 11, pp. 38-50, (1991)
Kim, J., P. Moin and R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, J. Fluid Mech., 177 (1987)
Launder, B. E. and J. L. Spalding, Mathematical models of turbulence, Academic Press, New York, (1972)
Launder, B. E., G. J. Reece and W. Rodi, Progress in the development of a Reynolds stress turbulence closure, J. Fluid Mech., 68 (1975) 537-566
Moin, P. and J. Kim, Numerical investigation of turbulent channel flow, J. Fluid, Mech., 18 (1982) 341-377
Murakami, S., A. Mochida and K. Hibi, Three-dimensional numerical simulation of airflow around a cubic model by means of large eddy simulation, J. Wind Eng. Ind. Aerodyn., 25 (1987) 291-305
Murakami, S. and A. Mochida, 3D numerical simulation of airflow around a cubic model by means of k-ε model, J. Wind Eng. Ind. Aerodyn., 31 (1988a) 283-303
Murakami, S. and A. Mochida, 3D numerical prediction of turbulent flow around buildings by means of k-e model, Building and Environment, 24 (1988b)
Murakami, S., Computational wind engineering, J. Wind Eng. Ind. Aerodyn., 36 (1990a) 517-538
Murakami, S., S. Kato and Y. Kondo, Examining k-ε EVM by means of ASM for a 3-D horizontal buoyant jet in enclosed space, Engineering Turbulence Modelling and Experiments, Elsevier Science Publishing, (1990b) 205-214
Murakami, S., A. Mochida and Y. Hayashi, Numerical simulation of velocity and diffusion field in an urban area, Energy and Buildings, 15-16 (1991a) 345-356
Murakami, S., A. Mochida, Y. Hayashi and S. Sakamoto, Numerical study on velocity-pressure field and wind forces for bluff bodies by k-e, ASM and LES, Preprints of 8th International Conference on Wind Engineering, (1991b)
Murakami, S., A. Mochida and Y. Hayashi, Scrutinizing k-ε EVM and ASM by means of LES and wind tunnel for flowfield around cube, 8th Symp. on Turbulent Shear Flows, (1991c)
Murakami, S., Kato, S, Utsumi, Y. and Mizutani, K., 2D room air flow simulation with processing (submitted), Kanto branch of AIJ Reserch Report (Tokyo), AIJ, (1992)
Rodi, W., ZAMM, 58 (1976) T219-T221
Schumann, U., Subgrid scale model for finite difference simulation of turbulent flows in plane channels and annuli, J. Comp. Phys., 18 (1975) 376-404
Smagorinsky, J. S., General circulation experiments with the primitive equations, Part1, Basic experiments, Mon. Weather Rev., 91 (1963) 99-164
Vickery, B. J., Fluctuating lift and drag on a long cylinder of square cross-section in a smooth and in a turbulent stream, J. Fluid Mech., 125 (1966) 481-494

Appendix 1 Model equations for the k-ε model

$$\frac{\partial \langle u_i\rangle}{\partial t}+\frac{\partial \langle u_i\rangle\langle u_j\rangle}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{\partial}{\partial x_j}\nu_t\left(\frac{\partial \langle u_i\rangle}{\partial x_j}+\frac{\partial \langle u_j\rangle}{\partial x_i}\right) \quad (1.1)$$

$$\frac{\partial \langle u_i\rangle}{\partial x_i}=0 \quad (1.2)$$

$$\frac{\partial k}{\partial t}+\frac{\partial k\langle u_j\rangle}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_j}\right)+\nu_t\left(\frac{\partial \langle u_i\rangle}{\partial x_j}+\frac{\partial \langle u_j\rangle}{\partial x_i}\right)\frac{\partial \langle u_i\rangle}{\partial x_j}-\varepsilon \quad (1.3)$$

$$\frac{\partial \varepsilon}{\partial t}+\frac{\partial \varepsilon\langle u_j\rangle}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x_j}\right)+\frac{\varepsilon}{k}\left(C_{\varepsilon 1}\nu_t\left(\frac{\partial \langle u_i\rangle}{\partial x_j}+\frac{\partial \langle u_j\rangle}{\partial x_i}\right)\frac{\partial \langle u_i\rangle}{\partial x_j}-C_{\varepsilon 2}\varepsilon\right) \quad (1.4)$$

$$\nu_t=C_\mu\frac{k^2}{\varepsilon} \quad (1.5)$$

C_μ : 0.09, σ_k : 1.0, σ_ε : 1.3,
$C_{\varepsilon 1}$: 1.44, $C_{\varepsilon 2}$: 1.92 (1.6)

Appendix 2 Model equations for RSM and ASM

$$\frac{\partial \langle u_i\rangle}{\partial t}+\frac{\partial \langle u_i\rangle\langle u_j\rangle}{\partial x_j}=-\frac{\partial p}{\partial x_i}-\frac{\partial \langle u_i'u_j'\rangle}{\partial x_j} \quad (2.1)$$

$$\frac{\partial \langle u_i\rangle}{\partial x_i}=0 \quad (2.2)$$

1) RSM

$$\frac{\partial \langle u_i'u_j'\rangle}{\partial t}+\frac{\partial \langle u_i'u_j'\rangle\langle u_j\rangle}{\partial x_j}=P_{ij}+D_{ij}+\Phi_{ij}-\varepsilon_{ij} \quad (2.3)$$

$$\frac{\partial \varepsilon}{\partial t}+\frac{\partial \varepsilon\langle u_j\rangle}{\partial x_j}=D_\varepsilon+\frac{\varepsilon}{k}(C_{\varepsilon 1}P_k-C_{\varepsilon 2}\varepsilon) \quad (2.4)$$

2) ASM

$$\frac{\partial k}{\partial t}+\frac{\partial k\langle u_j\rangle}{\partial x_j}=D_k+P_k-\varepsilon \quad (2.5)$$

$$\frac{\partial \varepsilon}{\partial t}+\frac{\partial \varepsilon\langle u_j\rangle}{\partial x_j}=D_\varepsilon+\frac{\varepsilon}{k}(C_{\varepsilon 1}P_k-C_{\varepsilon 2}\varepsilon) \quad (2.6)$$

$$(P_k-\varepsilon)\frac{\langle u_i'u_j'\rangle}{k}=P_{ij}+\Phi_{ij}-\varepsilon_{ij} \quad (2.7)$$

$$D_{ij}=\frac{\partial}{\partial x_m}\left(C_k\langle u_m'u_i'\rangle\cdot\frac{k}{\varepsilon}\cdot\frac{\partial \langle u_i'u_j'\rangle}{\partial x_l}\right) \quad (2.8)$$

$$D_k=\frac{\partial}{\partial x_m}\left(C_k\langle u_m'u_i'\rangle\cdot\frac{k}{\varepsilon}\cdot\frac{\partial k}{\partial x_l}\right) \quad (2.9)$$

$$D_\varepsilon=\frac{\partial}{\partial x_m}\left(C_\varepsilon\langle u_m'u_i'\rangle\cdot\frac{k}{\varepsilon}\cdot\frac{\partial \varepsilon}{\partial x_l}\right) \quad (2.10)$$

$$P_k=-\langle u_i'u_j'\rangle\frac{\partial \langle u_i\rangle}{\partial x_j} \quad (2.11)$$

$$P_{ij}=-\langle u_i'u_k'\rangle\frac{\partial \langle u_j\rangle}{\partial x_k}-\langle u_j'u_k'\rangle\frac{\partial \langle u_i\rangle}{\partial x_k} \quad (2.12)$$

$$\varepsilon_{ij}=\frac{2}{3}\cdot\delta_{ij}\varepsilon \quad (2.13)$$

$$\Phi_{ij}=\Phi_{ij(1)}+\Phi_{ij(2)}+\Phi_{ij(1)}^{w} \quad (2.14)$$

$$\Phi_{ij(1)}=-C_1\frac{\varepsilon}{k}\left(\langle u_i'u_j'\rangle-\frac{2}{3}\delta_{ij}k\right) \quad (2.15)$$

$$\Phi_{ij(2)}=-C_2\left(P_{ij}-\frac{2}{3}\delta_{ij}P_k\right) \quad (2.16)$$

$$\Phi_{ij(1)}^{w}=\sum_{w=1}^{w2}C_1'\frac{\varepsilon}{k}\left(\langle u_k'u_m'\rangle\cdot n_k^{(w)}\cdot n_m^{(w)}\cdot\delta_{ij}-\frac{3}{2}\langle u_k'u_i'\rangle\cdot n_k^{(w)}\cdot n_j^{(w)}-\frac{3}{2}\langle u_k'u_j'\rangle\cdot n_k^{(w)}\cdot n_i^{(w)}\right)\cdot\frac{k^{3/2}}{C_l\cdot h_n^{(w)}\varepsilon} \quad (2.17)$$

C_μ : 0.09, σ_k : 1.0, σ_ε : 1.3, C_1 : 1.8, C_2 : 0.6,
C_1' : 0.5, C_2' : 0.5, C_k : 0.22, C_ε : 0.16,
$C_{\varepsilon 1}$: 1.44, $C_{\varepsilon 2}$: 1.92, C_l : 2.5 (2.18)

Appendix 3 Model equations for LES

$$\frac{\partial \bar{u}_i}{\partial t}+\frac{\partial \bar{u}_i\bar{u}_j}{\partial x_j}=-\frac{\partial \bar{p}}{\partial x_i}+\frac{\partial}{\partial x_j}(\nu+\nu_{SGS})\left(\frac{\partial \bar{u}_i}{\partial x_j}+\frac{\partial \bar{u}_j}{\partial x_i}\right) \quad (3.1)$$

$$\frac{\partial \bar{u}_i}{\partial x_i}=0 \quad (3.2)$$

$$\nu_{SGS}=(C_s\bar{h})^2\left|\left(\frac{\partial \bar{u}_i}{\partial x_j}\left(\frac{\partial \bar{u}_i}{\partial x_j}+\frac{\partial \bar{u}_j}{\partial x_i}\right)\right)\right|^{1/2} \quad (3.3)$$

$$\bar{h}=(h_1h_2h_3)^{1/3} \quad (3.4)$$

C_s : 0.12 (3.5)

NOMENCLATURE

- x_i : three components of spatial coordinate (i = 1,2,3 : streamwise, lateral, vertical)
- $\langle f\rangle$: time-averaged value of f
- $\bar{f}$: filtered value of f
- f' : deviation from $\langle f\rangle$, $f'=f-\langle f\rangle$ (In LES, $f'=\bar{f}-\langle\bar{f}\rangle$)
- H_b : height of cube
- D : width of square prism
- u_i : three components of velocity vector
- u_b : u_1 value at inflow of computational domain at height H_b
- U_0 : u_1 value of uniform and smooth free stream
- p : pressure
- $\langle p_0\rangle$: reference static pressure
- ν_t : eddy viscosity
- ν_{SGS} : subgrid scale eddy viscosity
- k : turbulent kinetic energy, $k=\frac{1}{2}\langle u_i'u_i'\rangle$
- P_k : production of k
- ε : dissipation rate of k
- $\langle u_i'u_j'\rangle$: Reynolds stress
- P_{ij} : production of $\langle u_i'u_j'\rangle$
- ε_{ij} : dissipation rate of $\langle u_i'u_j'\rangle$
- C_{ij} : convection term of $\langle u_i'u_j'\rangle$
- D_{ij} : diffusion term of $\langle u_i'u_j'\rangle$
- Φ_{ij} : pressure-strain correlation term (consists here of $\Phi_{ij(1)}$, $\Phi_{ij(2)}$ and $\Phi_{ij(1)}^{w}$, cf. APPENDIX 2)
- h_P : mesh interval adjacent to solid wall
- $(u_i)_P$: tangential component of velocity vector at the near-wall node
- k_P : k value at the near-wall node
- ε_P : ε value at the near-wall node
- τ_w : wall shear stress
- $h_n^{(w)}$: vertical distance from the w-th wall
- C_P : instantaneous pressure coefficient, $C_P=(p-\langle p_s\rangle)/(\frac{1}{2}\rho\langle u_b\rangle^2)$
- $C_{P\,rms}$: rms pressure coefficient : $\sqrt{\langle p''^2\rangle}/(\frac{1}{2}\rho\langle u_b\rangle^2)$
- F_D : Instantaneous drag force
- F_L : Instantaneous lift force
- $S(n)$: power spectrum

When values are made dimensionless, representative length scale (H_b or D), velocity scale ($\langle u_b\rangle$ or U_0) and air density ρ are used.

MIYAGI NATIONAL COLLEGE
OF TECHNOLOGY

MEDESHIMA NATORI MIYAGI 981-12 JAPAN

1992.3.9

Dr. Martin Liddament
AIVC
Barclays Venture Centre
University of Warwick Science Park
Coventry CV4 7EZ
United Kingdom

Dear Martin,

Please find the enclosed papers entitled "The Future of CFD in Civil Engineering: Large-Scale Computation with Vector and Massive Parallel Computers" by Prof. Shuzo Murakami of Institute of Industrial Science, University of Tokyo. This paper describes the brief view of Computational Fluid Dynamics in future.

Please let me know if you have any question and I hope this can be any help of you.

Best Regards,

Yasuo Utsumi

Yasuo Utsumi
Department of Architecture
Miyagi National College of Technology
Natori, 981-12
Japan