Scientific Research Publishing

Open Journal of Fluid Dynamics, 2018, 8, 416-434
http://www.scirp.org/journal/ojfd
ISSN Online: 2165-3860
ISSN Print: 2165-3852

Large-Eddy Simulation of Airflow over a Steep, Three-Dimensional Isolated Hill with Multi-GPUs Computing

Takanori Uchida

Research Institute for Applied Mechanics (RIAM), Kyushu University, Fukuoka, Japan
Email: takanori@riam.kyushu-u.ac.jp

How to cite this paper: Uchida, T. (2018) Large-Eddy Simulation of Airflow over a Steep, Three-Dimensional Isolated Hill with Multi-GPUs Computing. *Open Journal of Fluid Dynamics*, **8**, 416-434.
https://doi.org/10.4236/ojfd.2018.84027

Received: November 29, 2017
Accepted: November 18, 2018
Published: November 21, 2018

Abstract

The present research attempted a Large-Eddy Simulation (LES) of airflow over a steep, three-dimensional isolated hill by using the latest multi-cores multi-CPUs systems. As a result, it was found that 1) turbulence simulations using approximately 50 million grid points are feasible and 2) the use of this system resulted in the achievement of a high computation speed, which exceeded the speed of parallel computation attained by a single CPU on one of the latest supercomputers. Furthermore, LES was conducted by using the multi-GPUs systems. The results of these simulations revealed the following findings: 1) the multi-GPUs environment which used the NVDIA® Tesla M2090 or the M2075 could simulate turbulence in a model with as many as approximately 50 million grid points. 2) The computation speed achieved by the multi-GPUs environments exceeded that by parallel computation which used four to six CPUs of one of the latest supercomputers.

Keywords

LES, Isolated Hill, Multi-Cores Multi-CPUs Computing, Multi-GPUs Computing

1. Introduction

Recently, the wind power industry has undergone rapid growth at an unprecedented rate across the world. This growth has been occurring because wind power generation has the best cost performance of all the renewable energies in terms of achieving a post-fossil fuel society and reducing CO_2 emissions. There is no doubt that wind power is the leading renewable energy even in Japan. The author is convinced that further dissemination of wind power generation will

contribute on the global scale to "green innovation", efforts to combat global warming. One technical issue which needs to be resolved in the near future in the field of wind power generation is to establish a numerical wind synopsis prediction technique which allows 1) accurate assessment of the local wind synopsis for wind turbines and 2) identification of local wind risks (terrain-induced turbulence) to wind turbines [1]-[15].

In our research group, the numerical wind synopsis prediction technique named the RIAM-COMPACT natural terrain version software has been developed [1]-[15]. The core technology of the RIAM-COMPACT is under continuous development at the Research Institute for Applied Mechanics (RIAM), Kyushu University. For the RIAM-COMPACT, which focuses on unsteady turbulence simulations, computation time had been an issue of concern. The present fluid simulation solver is compatible with multi-cores CPUs (Central Processing Units) such as the Intel Core i7 or Core i9, which has drastically reduced the computation time, leaving no appreciable problems in terms of the practical use of the RIAM-COMPACT software. Furthermore, the RIAM-COMPACT software has successfully been made compatible with GPGPU (General Purpose computing on GPU) computing. The concept of GPGPU is to widely apply the floating-point operation capacity of a GPU (Graphics Processing Unit) not only to graphics rendering but also to other numerical operations.

The present paper reports the results of the improvement in speed of the RIAM-COMPACT software using multi-cores CPUs, single and multi-GPUs for a flow field over a steep, three-dimensional isolated hill.

2. Summary of the RIAM-COMPACT Natural Terrain Version Software

In the present study, the RIAM-COMPACT natural terrain version software is used to numerically predict local airflows over complex terrain with high accuracy while avoiding numerical instability. The RIAM-COMPACT natural terrain version software uses collocated grids in a general curvilinear coordinate system. In these collocated grids, the velocity components and pressure are defined at the cell centers, and variables which result from the contravariant velocity components multiplied by the Jacobian are defined at the cell faces. As for the simulation technique, the Finite-Difference Method (FDM) is adopted, and a Large-Eddy Simulation (LES) model is used for the turbulence model. In LES, a spatial filter is applied to the flow field to separate eddies of various scales into Grid Scale (GS) components, which are larger than the computational grid cells, and Subgrid-Scale (SGS) components, which are smaller than the computational grid cells. Large-scale eddies, *i.e.* the GS components of turbulence eddies, are directly numerically simulated without relying on the use of a physically simplified model. On the other hand, the main effect of small-scale eddies, *i.e.* the SGS components, is to dissipate energy, and this dissipation is modeled based on the physical considerations of the SGS stress.

For the governing equations of the flow, a spatially-filtered continuity equation for incompressible fluid (Equation (1)) and a spatially-filtered Navier-Stokes equation (Equation (2)) are used:

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial \overline{u}_i}{\partial t} + u_j \frac{\partial \overline{u}_i}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \tag{2}$$

Supporting equations are given in Equations (3)-(8):

$$\tau_{ij} \approx \overline{u_i' u_j'} \approx \frac{1}{3} \overline{u_k' u_k'} \delta_{ij} - 2\nu_{\mathrm{SGS}} \overline{S}_{ij} \tag{3}$$

$$\nu_{\mathrm{SGS}} = \left(C_s f_s \Delta\right)^2 \left|\overline{S}\right| \tag{4}$$

$$\left|\overline{S}\right| = \left(2\overline{S}_{ij}\overline{S}_{ij}\right)^{1/2} \tag{5}$$

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) \tag{6}$$

$$f_s = 1 - \exp\left(-z^+/25\right) \tag{7}$$

$$\Delta = \left(h_x h_y h_z\right)^{1/3} \tag{8}$$

Because mean wind speeds of 8 - 10 m/s or higher are considered in the present study, the effect of vertical thermal stratification (density stratification), which is generally present in the atmosphere, is neglected. Furthermore, the effect of surface roughness is included by reconstructing the irregularities of the terrain surface in high resolution.

The computational algorithm and the time marching method are based on a Fractional-Step (FS) method [16] and the Euler explicit method, respectively. The Poisson's equation for pressure is solved by the Successive Over-Relaxation (SOR) method. For discretization of all the spatial terms except for the convective term in Equation (2), a second-order central difference scheme is applied. For the convective term, a third-order upwind difference scheme is applied. An interpolation technique based on 4-point differencing and 4-point interpolation by Kajishima [17] is used for the fourth-order central differencing that appears in the discretized form of the convective term. In the weighting of the numerical diffusion term of the third-order upwind differencing of the convective term, α = 3.0 is commonly applied in the Kawamura-Kuwahara Scheme [18]. However, α = 0.5 is used in the present study to minimize the influence of numerical diffusion. For LES subgrid-scale modeling, the commonly used Smagorinsky model [19] is adopted. Additionally, a wall-damping function is used with a model coefficient of 0.1.

3. Flow field and Simulation Set-Up for the Present Study

In the present study, the required computation time is compared among various

hardware configurations for a flow field over a steep, three-dimensional isolated hill. The geometry of the isolated hill is represented by the following function:

$$z(r) = 0.5\mathrm{h} \times \{1 + \cos(\pi r/a)\}, r = \left(x^2 + y^2\right)^{1/2}, a = 2\mathrm{h} \quad (9)$$

where x, y, and z are the streamwise, spanwise, and vertical directions, respectively. In addition, h is the hill height, and a is the radius of the hill at its base.

Figure 1 illustrates the computational domain, the coordinate system, and the boundary conditions used in the simulation. The spatial size of the computational domain is 5,601 h × 9 h × 10 h in the x, y, and z directions, respectively. The number of grid points is 5,821 × 121 × 71 in the x, y, and z directions, respectively (total: approximately 50 million grid points; the memory used is approximately 15 GB) (see Table 1). Furthermore, we compared the calculation results using 2.23 million grid points and 50 million grid points shown in the Appendix, and confirmed beforehand that there was no significant difference in the calculation results.

Figure 2 shows the computational grid in the vicinity of the isolated hill. The grid spacing is set unevenly between 0.04 h and h in the x-direction, between 0.05 h and 0.5 h in the y-direction, and between 0.0035 h and 0.5 h in the z-direction.

Regarding the boundary conditions, uniform inflow conditions are applied at the inflow boundary, slip conditions are applied at the side and upper boundaries, and convective outflow conditions are applied at the outflow boundary. At the ground surface, non-slip boundary conditions are applied. The Reynolds number is set to Re (=Uh/υ) = 10^4, where h is the hill height and U is the wind speed at height h at the inflow boundary. The time step is set to $\Delta t = 2 \times 10^{-3}$ h/U.

Subsequently, the pattern of the flow that forms around the isolated hill investigated in the present study is discussed (see Figure 3). In the wind tunnel experiment shown in Figure 3(a), the flow field is visualized with the smoke wire technique as described below. Several wires (0.3 mm nichrome wires) are laid out at various heights immediately upstream of the model of the hill parallel to one another. After a mixture of liquid paraffin and aluminum powder is applied

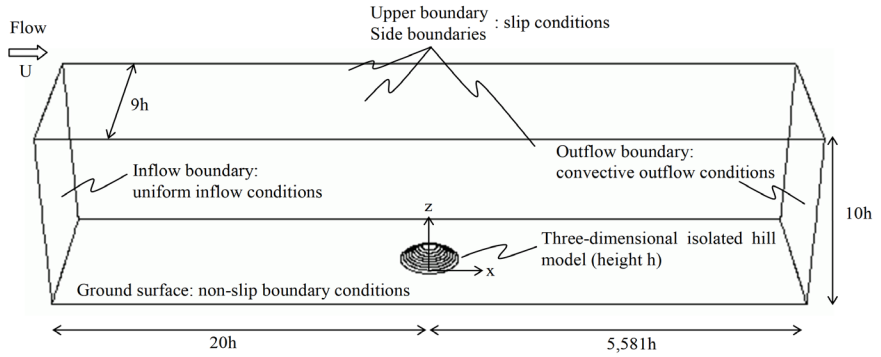


Figure 1. Computational domain, coordinate system, and boundary conditions.

Table 1. Number of computational grid points, memory used, and number of calculation steps.

Number of computational grid points	5,821 × 121 × 71 points (approx. 50 million points)
Memory used	Approx. 15 GB
Number of calculation steps	5,000

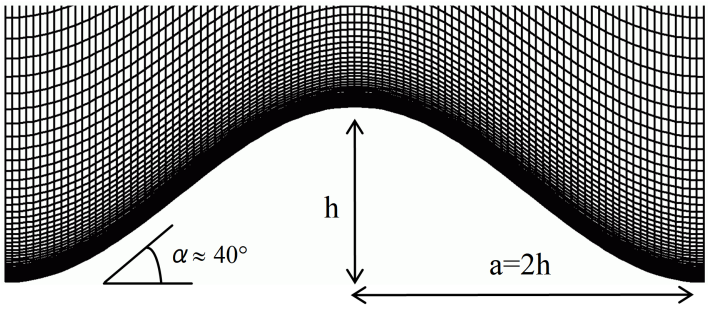


Figure 2. Computational grid in the vicinity of the isolated hill on the central plane (y = 0) aligned with the streamwise direction (x).

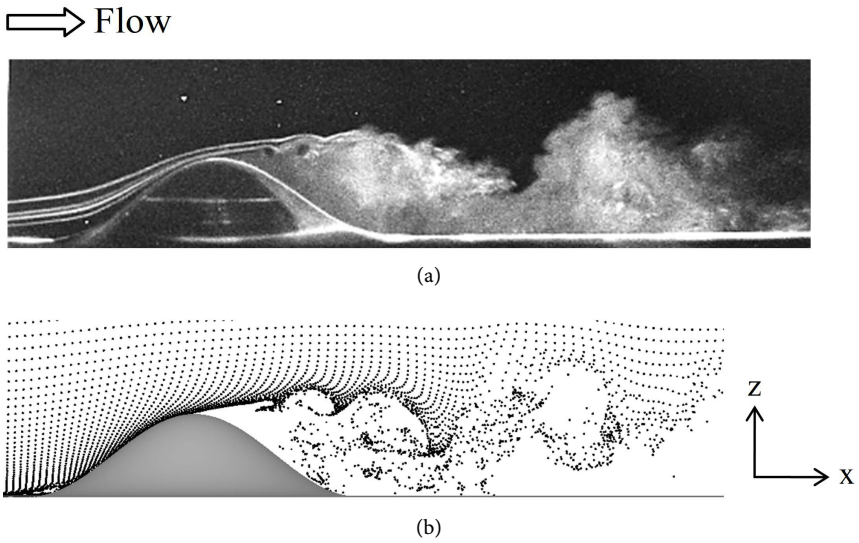


Figure 3. Visualization of flow in the vicinity of an isolated hill. Instantaneous field. (a) Wind tunnel experiment, smoke wire technique; (b) Numerical simulation (LES), passive particle tracking method.

to these wires, an electric current is introduced, which vaporizes the mixture into smoke and allows the visualization of the flow field. Several 1 kW projectors with slit apertures are installed in the upper part of the wind tunnel. With the light from these projectors, the flow in the central plane (y = 0) aligned with the streamwise (x) direction is visualized. A standard lens is used for camera-based photography. The aperture is set to f1.2, and the shutter speed (exposure time) is set to 1/125 s. Furthermore, the wind speed at height h at the inflow boundary is set to 1.5 m/s, and the flow conditions in the wind tunnel are made identical to

those from the numerical simulation by the RIAM-COMPACT natural terrain version software ($Re = 10^4$). In order to closely examine the behavior of the flow, particularly that of the boundary layer which separates in the vicinity of the top of the isolated hill (*i.e.*, separated shear layer), the wire heights are adjusted so that the smoke drifts near the surface of the isolated hill.

On the other hand, in the numerical simulation by the RIAM-COMPACT natural terrain version software as shown in **Figure 3(b)**, the flow field is visualized using the passive particle tracking method. The time interval for releasing passive particles is set to $\Delta t = 0.1$ h/U. The result shown in **Figure 3(b)** is that from a total of 100 frames (from the time t = 50 h/U to 60 h/U). The qualitative behaviors of the flows from the numerical simulation and the wind tunnel experiment are quite similar to each other. More specifically, the flow separates in the vicinity of the top of the isolated hill, and this separated shear layer rolls into isolated eddies. The isolated eddies coalesce into large-scale eddies, which are then shed downstream of the isolated hill. The comparison between RANS [20] results and the present LES results are summarized in the latest article [7], and the prediction accuracy of the present LES approach by comparison with wind tunnel experiments is discussed in the article [14].

4. Summary of the Computing Environments Used for the Present Study and Computation Speed of Multi-Cores CPUs

In this section, the computing environments used for the present study will be described. **Figure 4** and **Figure 5** show examples of systems equipped with Sandy Bridge-EP CPUs. Furthermore, **Table 2** shows the specifications of the Sandy Bridge-EP system used in the present study. **Figure 6** gives a brief overview of the vector computer used in the present study for comparisons of the computation speed. This NEC SX-9F was a general purpose computer owned by the Research Institute for Applied Mechanics (RIAM), Kyushu University, and finished operating in January 2015.

In the present study, the flow field that forms around the isolated hill at time t = 50 h/U is adopted as the initial flow condition. Starting at this time, an additional 5,000 calculation steps (t = 50 h/U to 60 h/U) are executed. For this simulation, a computational grid with 50 million points is used. The time required for these calculation steps by the Sandy Bridge-EP CPUs is compared to that by the supercomputer, SX-9F. The results of this comparison are shown in **Figure 7**. **Figure 7** also includes the computation speeds of the Intel Xeon X5600 series for comparison. The results listed as "Auto" in the table correspond to those from a

Table 2. Specifications of the Sandy Bridge-EP system used in the present study.

CPUs:	Intel Xeon E5-2670 @ 2.6 GHz × 2 (16 cores)
Memory:	48 GB
OS:	RHEL 6.1
Compiler:	Intel Composer XE 12.0.4

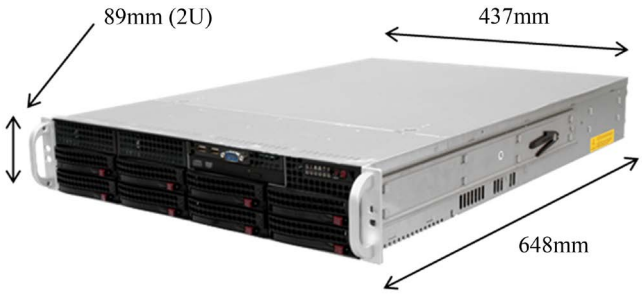


Figure 4. Example system 1, rack mount model.

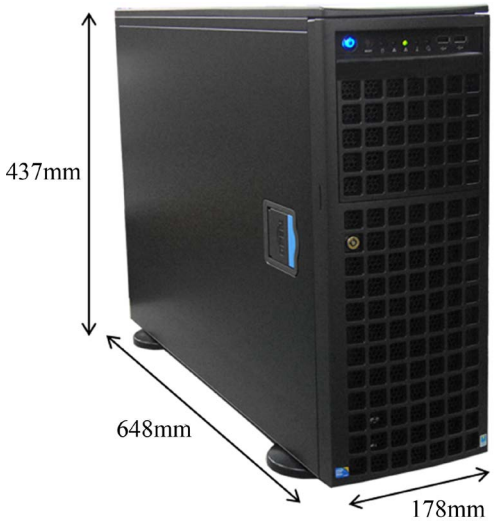
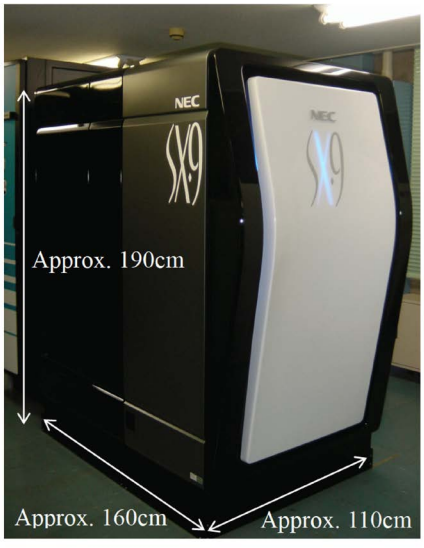
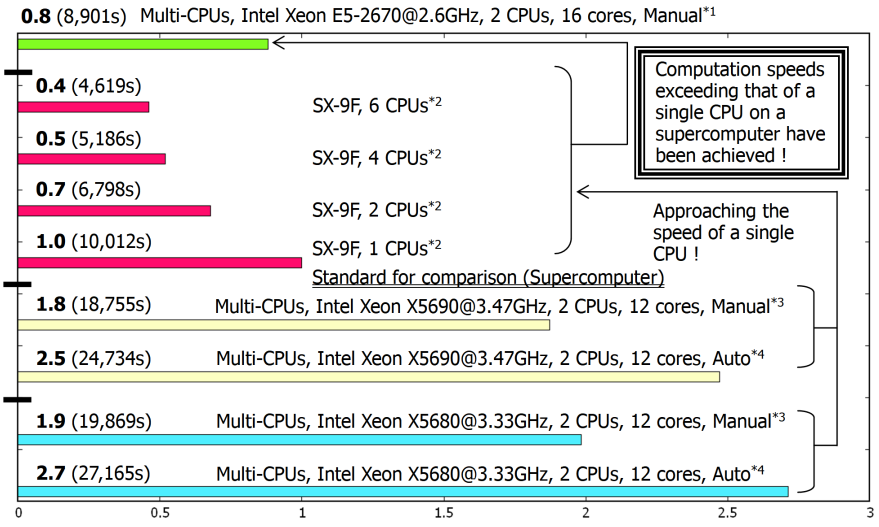


Figure 5. Example system 2, workstation model.



[Configuration]
Number of CPUs: 6 (92.16 GFLOPS / unit × 6 = 552.96 GFLOPS)
Memory: 256 GB
External disk drive: iStorage D3-10 4TB (RAID5)

Figure 6. NEC SX-9F used in the present study (a vector computer, released in 2007).



*1 The "-assume buffered_io -ipo -O3 -no-prec-div -xAVX -fpp –openmp" option for Intel Composer XE 12.0.4 is used.
→Vector operations are optimized by AVX.
Upon execution, memory allocation is also optimized using "/usr/bin/time numactl --interleave=all <sequence of commands...>"
*2 The "-Pauto" option for FORTRAN90/SX is used.
*3 The "/fast /Qopenmp /fpp" option for Intel Composer XE 2011 SP1 Update 8 is used.
*4 The "/fast / Qparallel" option for Intel Composer XE 2011 SP1 Update 8 is used.

Figure 7. Comparison of computation speeds for simulations using 50 million grid points. Memory used: approximately 15 GB. (Bold numbers indicate the speed ratio with respect to a single CPU on the SX-9F. Numbers inside parentheses represent the actual computation time.)

parallel computation (OpenMP) which uses the auto-parallelization option "/Qparallel" of the Intel Fortran compiler. On the other hand, the results listed as "Manual" are those from cases in which the parallelization directives for OpenMP are inserted in the computation program in advance and these directives are enabled by specifying the "/Qopenmp" option for the Intel Fortran compiler. An examination of the results of the parallel computation by the Intel Xeon X5600 series reveals that the computation speeds are higher in the cases of "Manual" than those in the cases of "Auto" by 20% to 30%. In both cases, the performance of the Intel Xeon X5600 series is approaching that of a single CPU on a supercomputer.

In contrast, the "manual" simulation with processors from the ultra high-end Sandy Bridge-EP Xeon E5-2600 CPU series successfully outperforms a single CPU on the supercomputer, resulting in the achievement of high-speed computation. For the simulation with Sandy Bridge-EP CPUs, interleaved memory is used. Interleaved memory enables parallel access to multiple memory modules, thus it is an effective function to retrieve a large volume of continuous data at once.

5. Summary of the Computing Environments Used for the Present Study and Computation Speed of Multi-GPUs Computing

In this section, the computing environments used for the present study will be described. Figures 8-10 show the computer models equipped with GPUs used

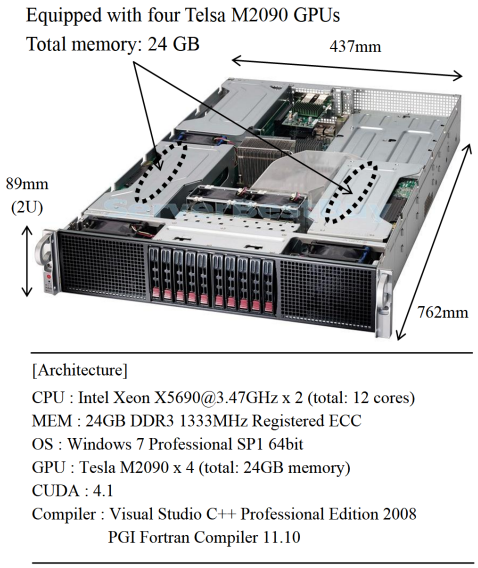


Figure 8. GPGPU rackmount server used in the present study.

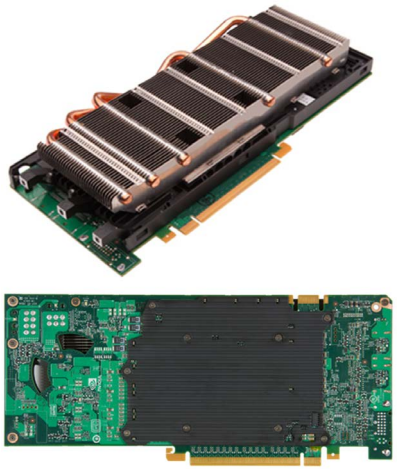


Figure 9. GPU used in the present study: Tesla M2090 (512 CUDA core, 6 GB memory).

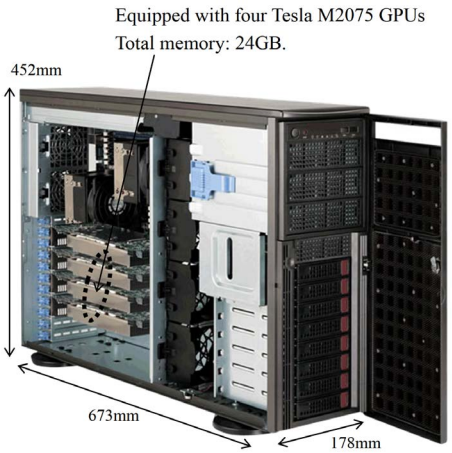
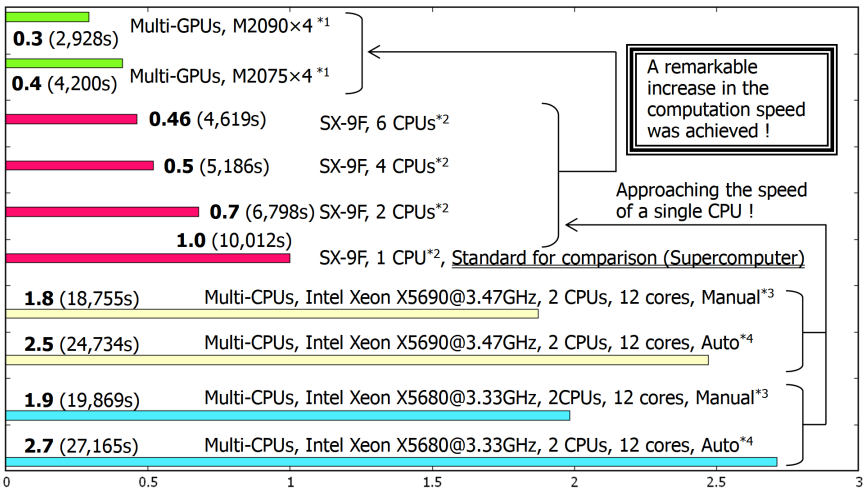


Figure 10. GPGPU workstation used in the present study, Tesla M2075 (448 CUDA core, memory: 6 GB) × 4.

in the study. The computer model shown in Figure 8 is equipped with four NVIDIA® Tesla M2090 6GB (512 CUDA core) GPUs (see Figure 9). The computer model shown in Figure 10 is equipped with four NVIDIA® Tesla M2075 6GB (448 CUDA core) GPUs.

In the present study, the flow field which forms around the isolated hill at time t = 50 h/U is adopted as the initial flow condition. Starting at this time, an additional 5,000 calculation steps (t = 50 h/U to 60 h/U) are executed, and the time required for these calculation steps is compared among various hardware configurations. Figure 11 shows the results obtained from this analysis. Figure 11 also shows the results from the latest multi-CPUs systems for comparison. The results of the multi-CPUs systems listed as "Auto" indicate those from parallel computations (OpenMP) which use the auto-parallelization option "/Qparallel" of the Intel Fortran compiler. On the other hand, the results listed as "Manual" indicate those in which the parallelization directives for OpenMP are inserted in the computation program in advance and these directives are enabled by specifying the "/Qopenmp" option of the Intel Fortran compiler. An examination of the obtained results reveals that the computation speeds of the "Manual" cases are approximately 20% to 30% higher than those of the "Auto" cases. It is clear that the computation speed of both cases is close to that of a single CPU in a supercomputer. As of the time of writing, Intel's new ultra high-end CPU "Sandy Bridge-E" has been released, which makes further improvement in the computation speed promising (see Figure 7).

Subsequently, the results from the multi-GPUs systems are examined. Here,



*1 A code which is written using a combination of CUDA C and Fortran and which is compatible with multi-GPUs systems is used. ECC (Error Checking and Correcting) setting = OFF. The NVIDIA CUDA compiler is used for the part of the code written in CUDA. A PGI compiler is used for the part of the code written in Fortran.
*2 The "-Pauto" option for FORTRAN90/SX is used.
*3 The "/fast/Qopenmp/fpp" option for Intel Composer XE 2011 SP1 Update 8 is used.
*4 The "/fast /Qparallel" option for Intel Composer XE 2011 SP1 Update 8 is used.

Figure 11. Comparison of computation speeds for simulations using 50 million grid points. Memory used: approximately 15 GB. (Bold numbers indicate the speed ratio with respect to a single CPU on the SX-9F. Numbers inside parentheses represent the actual computation time.)

the author wishes to reemphasize the following point. The previous research discussed the computation speed of these single GPUs: the NVIDIA® Tesla C2050 (448 CUDA cores) and the NVIDIA® GeForce GTX 580 (512 CUDA cores). However, with both GPUs, simulations using more than 8 million points were not feasible as the VRAM (Video Random Access Memory) capacities of the GPUs were 3GB. Recently, turbulence simulations using approximately 50 million grid points has become possible in a multi-GPUs environment as a result of the release of a new NVIDA GPU, upgrades to CUDA (Compute Unified Device Architecture) and the resulting achievement of inter-GPUhigh-speed communications via GPU Direct.

A remarkably large increase in the computation speed has been achieved with both the machine equipped with four NVIDIA® Tesla M2090 6GB (512 CUDA cores) GPUs and the machine equipped with four NVIDIA® Tesla M2075 6GB (448 CUDA cores) GPUs. The computation speeds achieved in both cases exceed that of the parallel computation which uses six CPUs on the SX-9F supercomputer.

6. Summary

The present research attempted a Large-Eddy Simulation (LES) of airflow over a steep, three-dimensional isolated hill by using the latest multi-cores multi-CPUs systems. As a result, it was found that 1) turbulence simulations using approximately 50 million grid points are feasible and 2) the use of this system resulted in the achievement of a high computation speed, which exceeded the speed of parallel computation attained by a single CPU on one of the latest supercomputers. Furthermore, LES was conducted by using the multi-GPUs systems. The results of these simulations revealed the following findings: 1) the multi-GPUs environment which used the NVDIA® Tesla M2090 or the M2075 could simulate turbulence in a model with as many as approximately 50 million grid points. 2) The computation speed achieved by the multi-GPUs environments exceeded that by parallel computation which used four to six CPUs of one of the latest supercomputers.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 17H02053. In addition, the author has received considerable assistance from HPC SYSTEMS, Inc. (http://www.hpc.co.jp/) in various matters including evaluation of computation time. The author wishes to express his gratitude to HPC SYSTEMS, Inc.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Uchida, T. (2018) Design Wind Speed Evaluation Technique in Wind Turbine In-

stallation Point by Using the Meteorological and CFD Models. *Journal of Flow Control, Measurement & Visualization*, **6**, 168-184. https://doi.org/10.4236/jfcmv.2018.63014

[2] Uchida, T. (2018) Computational Investigation of the Causes of Wind Turbine Blade Damage at Japan's Wind Farm in Complex Terrain. *Journal of Flow Control, Measurement & Visualization*, **6**, 152-167. https://doi.org/10.4236/jfcmv.2018.63013

[3] Uchida, T. (2018) Numerical Investigation of Terrain-Induced Turbulence in Complex Terrain by Large-Eddy Simulation (LES) Technique. *Energies*, **11**, 2638. https://doi.org/10.3390/en11102638

[4] Uchida, T. (2018) Computational Fluid Dynamics Approach to Predict the Actual Wind Speed over Complex Terrain. *Energies*, **11**, 1694. https://doi.org/10.3390/en11071694

[5] Uchida, T. (2018) LES Investigation of Terrain-Induced Turbulence in Complex Terrain and Economic Effects of Wind Turbine Control. *Energies*, **11**, 1530. https://doi.org/10.3390/en11061530

[6] Uchida, T. (2018) Computational Fluid Dynamics (CFD) Investigation of Wind Turbine Nacelle Separation Accident over Complex Terrain in Japan. *Energies*, **11**, 1485. https://doi.org/10.3390/en11061485

[7] Uchida, T. and Li, G. (2018) Comparison of RANS and LES in the Prediction of Airflow Field over Steep Complex Terrain. *Open Journal of Fluid Dynamics*, **8**, 286-307. https://doi.org/10.4236/ojfd.2018.83018

[8] Uchida, T. (2018) Large-Eddy Simulation and Wind Tunnel Experiment of Airflow over Bolund Hill. *Open Journal of Fluid Dynamics*, **8**, 30-43. https://doi.org/10.4236/ojfd.2018.81003

[9] Uchida, T. (2017) High-Resolution LES of Terrain-Induced Turbulence around Wind Turbine Generators by Using Turbulent Inflow Boundary Conditions. *Open Journal of Fluid Dynamics*, **7**, 511-524. https://doi.org/10.4236/ojfd.2017.74035

[10] Uchida, T. (2017) High-Resolution Micro-Siting Technique for Large Scale Wind Farm Outside of Japan Using LES Turbulence Model. *Energy and Power Engineering*, **9**, 802-813. https://doi.org/10.4236/epe.2017.912050

[11] Uchida, T. and Ohya, Y. (2011) Latest Developments in Numerical Wind Synopsis Prediction Using the RIAM-COMPACT CFD Model-Design Wind Speed Evaluation and Wind Risk (Terrain-Induced Turbulence) Diagnostics in Japan. *Energies*, **4**, 458-474. https://doi.org/10.3390/en4030458

[12] Uchida, T., Ohya, Y. and Sugitani, K. (2011) Comparisons between the Wake of a Wind Turbine Generator Operated at Optimal Tip Speed Ratio and the Wake of a Stationary Disk. *Modelling and Simulation in Engineering*, **2011**, Article ID: 749421. https://doi.org/10.1155/2011/749421

[13] Uchida, T., Maruyama, T. and Ohya, Y. (2011) New Evaluation Technique for WTG Design Wind Speed using a CFD-Model-Based Unsteady Flow Simulation with Wind Direction Changes. *Modelling and Simulation in Engineering*, **2011**, Article ID: 941870. https://doi.org/10.1155/2011/941870

[14] Uchida, T. and Ohya, Y. (2008) Micro-Siting Technique for Wind Turbine Generators by Using Large-Eddy Simulation. *Journal of Wind Engineering and Industrial Aerodynamics*, **96**, 2121-2138. https://doi.org/10.1016/j.jweia.2008.02.047

[15] Uchida, T. and Ohya, Y. (2001) Large-Eddy Simulation of Turbulent Airflow over Complex Terrain. *Journal of Wind Engineering and Industrial Aerodynamics*, **91**, 219-229. https://doi.org/10.1016/S0167-6105(02)00347-1

[16] Kim, J. and Moin, P. (1985) Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, **59**, 308-323. https://doi.org/10.1016/0021-9991(85)90148-2

[17] Kajishima, T. (1994) Upstream-Shifted Interpolation Method for Numerical Simulation of Incompressible Flows. *Bull. Jpn. Soc. Mech. Eng. B*, **60**, 3319-3326. (In Japanese) https://doi.org/10.1299/kikaib.60.3319

[18] Kawamura, T., Takami, H. and Kuwahara, K. (1986) Computation of High Reynolds Number Flow around a Circular Cylinder with Surface Roughness. *Fluid Dynamics Research*, **1**, 145-162. https://doi.org/10.1016/0169-5983(86)90014-6

[19] Smagorinsky, J. (1963) General Circulation Experiments with the Primitive Equations, Part 1, Basic Experiments. *Monthly Weather Review*, **91**, 99-164. https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2

[20] Prospathopoulos, J.M., Politis, E.S. and Chaviaropoulos, P.K. (2012) Application of a 3D RANS Solver on the Complex Hill of Bolund and Assessment of the Wind Flow Predictions. *Journal of Wind Engineering and Industrial Aerodynamics*, **107**, 149-159. https://doi.org/10.1016/j.jweia.2012.04.011

Appendix

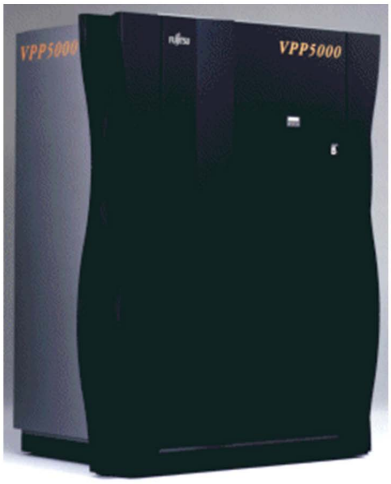
In this section, the results of the improvement in speed of the numerical wind synopsis prediction technique RIAM-COMPACT using multi-cores CPUs and single GPUs. First, the computing environment used for the present study will be described. Figure 12 and Figure 13 show the computing environment used for the GPU testing. The GPU on the HP Z800 workstation is an NVIDIA® Tesla™ C2050 3 GB (see Figure 13). In order to create an execution binary to be run on Windows for the GPU, the PGI CUDA Fortran compiler (v.11.4), which was developed jointly by PGI (Portland Group, Inc.) and NVIDIA Corporation, is installed. Intel Composer XE 2011 (Update 3) is also installed to evaluate the performance of the computing environment using multi-cores CPUs. Figure 14 gives summaries of the vector computers used in the present study. This computer is general purpose computers owned by the Research Institute for Applied Mechanics (RIAM), Kyushu University.

The spatial size of the computational domain is 40 h (±20 h) × 9 h × 10 h in the x, y, and z directions, respectively. The number of grid points is 260 × 121 × 71 in the x, y, and z directions, respectively (total: approximately 2.23 million grid points; the memory used is approximately 800 MB), h is the hill height. Regarding the boundary conditions, the wind velocity profile applied at the inflow boundary is based on a commonly used empirical power law. A power law index

[Configuration]
CPU: Intel Xeon X5680 (3.33 GHz, 6 cores) × 2 (Total : 12 cores)
GPU : NVIDIA Tesla C2050 × 1 (3 GB)
Memory: 48 GB DDR-3 SDRAM
OS: Windows 7 64-bit version
GPU compiler: PGI Accelerator Fortran/C/C++ WS for Windows 32/64bit (v.11.4)
CPU compiler: Intel Composer XE 2011 (Update 3)
CUDA (Compute Unified Device Architecture): CUDA 3.2

Figure 12. HP Z800 Workstation used in the present study.

Figure 13. GPU used in the present study: NVIDIA Tesla C2050 (3 GB memory).



[Configuration]
Number of PE: 1 (9.6GFLOPS)
Memory: 4 GB

Figure 14. FUJITSU VPP5000U used in the present study (a vector computer, released in 1999).

is set to 5. Furthermore, slip conditions are applied at the side and upper boundaries and convective outflow conditions are applied at the outflow boundary. At the ground surface, non-slip boundary conditions are applied. The Reynolds number is set to Re (=Uh/υ) = 10^4, where h is the hill height and U is the wind speed at height h at the inflow boundary. The time step is set to $\Delta t = 2 \times 10^{-3}$ h/U. In the present study, the flow field that forms around the isolated hill at time t = 100 h/U is adopted as the initial flow condition. Starting at this time, an additional 5,000 calculation steps (t = 100 h/U to 110 h/U) are executed, and the time required for these calculation steps is compared among various hardware configurations.

Table 3 shows the results obtained from the use of an HP Z800 Workstation owned by our research group. The results under "Auto" in the table correspond to those from a parallel computation (OpenMP) which uses the auto-parallelization option "/Qparallel" of the Intel Fortran compiler. On the other hand, the results

Table 3. Results obtained from HP Z800 Workstation for the case which uses approximately 2.23 million grid points.

Hardware architecture	OS	Single core CPU*1	12 core CPU		Single GPU*4
			Auto*2	Manual*3	
CPU: Intel Xeon X5680 (3.33 GHz, 6 cores) × 2 (total: 12 cores) "Westmere" GPU: NVIDIA Tesla C2050 × 1 (3 GB)	Windows 7 (64 bit)	3727.93 s	1165.38 s	902.61 s	391.74 s
		Ratio of speed improvement	×3.20 (vs. the single core CPU)	×4.13 (vs. the single core CPU)	×9.52 (vs. the single core CPU)
		×0.56*5	×1.79*5	×2.31*5	×5.33*5

*1Intel Fortran compiler option "/fast" is used; *2Intel Fortran compiler option "/fast/Qparallel" is used; *3Intel Fortran compiler option "/fast/Qopenmp/fpp" is used; *4PGI compiler option "-fastsse -ta = nvidia, cuda3.2, cc2.0" (used with the GPU) is used. ECC (Error Checking and Correcting) setting = OFF; *5Compared to the result from the case which uses a single PE of a FUJITSU VPP5000U (2088.99s).

under "Manual" are those from the case in which the parallelization directives are inserted in the computation program in advance and these directives are enabled by specifying the "/Qopenmp" option of the Intel Fortran compiler. First, a close examination of the results of the parallel computation by the CPUs reveals that the value of the ratio of the speed improvement is higher in the case of "Manual" than that in the case of "Auto" as is expected. In both cases, the ratio of the speed improvement is about three to four based on a comparison against the computation speed of a single core CPU, suggesting that the capacity of the 12 core CPUs is higher than that of a single PE of a supercomputer from a few years ago.

Next, the results from the case which uses a GPU are examined. The NVIDIA Tesla C2050 is equipped with an ECC (Error Checking and Correcting) function, which detects and corrects wrong values stored in the GPU's memory. In the present performance evaluation, it is confirmed that 1) no difference arises in the computational results from the on/off status of the ECC function and 2) the computation speed is higher in the case with the ECC setting = OFF than in the case with the ECC setting = ON. Although the results under consideration are obtained from the use of only a single GPU, it is evident that the ratio of speed improvement is remarkably large: approximately 9 based on a comparison against the computation speed achieved using a single core CPU. The computation speed of the GPU is equivalent to that obtained from a parallel computation which uses 4 to 6 CPUs on the latest supercomputer (see **Table 6**).

In the present study, with the cooperation of HPC SYSTEMS, Inc., a performance evaluation is also made on an HPC SYSTEMS, Inc. computing environment equipped with the currently most up-to-date CPU (Intel Core i7 2600 K, 3.4 GH, Sandy Bridge). A new enhanced instruction set called Intel AVX (Advanced Vector eXtensions) has been added to Sandy Bridge. This is a new SIMD (vector operation) instruction set for an Intel X86 CPU and succeeds SSE (Streaming SIMD Extensions). Intel AVX is a technology which speeds up floating-point operations, and thus is expected to reduce the computation time for fluid simulations. In the present study, the performance of Intel AVX is also examined. The system of HPC SYSTEMS, Inc. is also equipped with the

NVIDIA GeForce GTX580 GPU (see Figure 15). Although the GTX580 has no ECC function, the price of the GTX580 is only about 1/6 that of the Tesla C2050. Thus, it can be said that the GTX580 is a quite powerful GPU for practical use. An examination of the results for the CPUs given in Table 8 and Table 6 reveals that the computation time improves by approximately 20% with the use of the Intel AVX technology. Regarding the GPU, the computation speed of the NVIDIA GeForce GTX580 is approximately 20% faster than the computation speed of the NVIDIA Tesla C2050 (see Table 4 and Table 5). For reference, Table 6 and Table 7 show the results obtained from the NEC SX-9F vector computer, which was used for the comparisons of the computation time of multi-cores CPUs and single GPUs.

Table 8 shows the study results from the simulations conducted on an HP Z800 Workstation in which the number of grid points is varied. The performance evaluation in the present study leads to the confirmation that simulations with up to 8 million grid points can be executed on the NVIDIA Tesla C2050 (3 GB memory). However, an execution of the simulation with 10 million grid points is not possible because of insufficient memory capacity. Table 8 shows that the computation speed of the GPU is approximately 8 times faster than the computation speed of the singlecore CPU even in cases with different numbers of grid points.

Table 4. Results for the case which uses approximately 2.23 million grid points.

Hardware architecture	OS	Single core CPU*1	4 core CPU	Single GPU*3
			Manual*2	
CPU: Intel Core i7 2600 K (3.4 GHz)× 1 "Sandy Bridge" GPU : NVIDIA GeForce GTX580 × 1 (1.5 GB)	Windows 7 (64 bit)	2625 s	1288 s	310 s
		Ratio of speed improvement	×2.04 (vs. the single core CPU)	×8.47 (vs. the single core CPU)
		×0.80*4	×1.62*4	×6.74*4

*1Intel Fortran compiler option "/fast" is used; *2Intel Fortran compiler option "/fast/Qopenmp /fpp" is used; *3PGI compiler option "-fastsse -ta = nvidia, cuda3.2, cc2.0 -Mpreprocess" (used with the GPU) is used. ECC (Error Checking and Correcting) setting = OFF; *4Compared to the result from the case which uses a single PE of a FUJITSU VPP5000U (2088.99s).

Table 5. Resultsfor the case which uses approximately 2.23 million grid points. Same as Table 4 except with the use of AVX.

Hardware architecture	OS	Single core CPU*1	4 core CPU	Single GPU*3
			Manual*2	
CPU: Intel Core i7 2600K (3.4 GHz)× 1 "Sandy Bridge" *With the use of AVX (Advanced Vector eXtensions) A new SIMD (vector operation) instruction set for an Intel X86 CPU, succeeding SSE. GPU: NVIDIA GeForce GTX580 × 1 (1.5 GB)	Windows 7 (64bit)	2208 s	1193 s	310 s
		Ratio of speed improvement	×1.85 (vs. the single core CPU)	×7.12 (vs. the single core CPU)
		×0.95*4	×1.75*4	×6.74*4

*1Intel Fortran compiler option "/fast/QxAVX" is used; *2Intel Fortran compiler option "/fast/Qopenmp /fpp/QxAVX" is used; *3PGI compiler option "-fastsse -ta = nvidia, cuda3.2, cc2.0 -Mpreprocess" (used with the GPU) is used. ECC (Error Checking and Correcting) setting = OFF; *4Compared to the result from the case which uses a single PE of a FUJITSU VPP5000U (2088.99s).

Table 6. Results obtained from NEC SX-9F (vector computer) for the case which uses approximately 2.23 million grid points.

Hardware architecture	1 CPU*[1]	2 CPUs*[1]	4 CPUs*[1]	6 CPUs*[1]
Number of CPUs: 6 (92.16GFLOPS)	709.28 s	469.49 s	344.46 s	302.42 s
	Ratio of speed improvement	×1.51 (vs 1 CPU)	×2.06 (vs 1 CPU)	×2.35 (vs 1 CPU)
	×2.94*[2]	×4.45*[2]	×6.06*[2]	×6.91*[2]

*[1]Fortran compiler option "-Pauto" is used; *[2]Compared to the result from the case which uses a single PE of a FUJITSU VPP5000U (2088.99s).

Table 7. Results obtained from NEC SX-9F (vector computer) for the cases which use approximately 4, 6, 8, and 10 million grid points.

Number of grid points	4 million*[1]	6 million*[1]	8 million*[1]	10 million*[1]
Number of CPUs: 6	455.93s	682.00s	893.53s	1097.26s
Ratio of speed improvement compared to the case in which 2.23 million grid points are used:302.42 s	×1.51	×2.26	×2.95	×3.63

*[1]Fortran compiler option "-Pauto" is used.

Table 8. Results obtained from HP Z800 Workstation for the cases which use approximately 4, 6, and 8 million grid points.

Number of grid points	Approx. 4 million	Approx. 6 million	Approx. 8 million
Single core CPU	6007 s	9731 s	12,768 s
GPU *ECC disabled	759 s	1189 s	1599 s
Ratio of speed improvement	×7.91 (vs. the single core CPU)	×8.18 (vs. the single core CPU)	×7.98 (vs. the single core CPU)

Figure 15. GPU used in the present study: NVIDIA GeForce GTX580 (1.5 GB memory).

The present study investigated the improvement of the speed of the numerical wind synopsis prediction technique RIAM-COMPACT with the use of mul-

ti-cores CPUs and single GPUs. As a result, it was found that 1) numerical fluid simulations using on the order of 8 million grid points are possible with the use of a GPU, specifically, the NVIDIA Tesla C2050 and the GeForce GTX580 and 2) computation on the GPUs used in the study remarkably increases the computation speed relative to computation on the singlecore CPUs used in the present study by a factor of approximately eight. It is especially worth pointing out that the computation speed improvement achieved by NVIDIA Tesla C2050 and the GeForce GTX580 for the simulation based on approximately 2.23 million grid points was equivalent to that obtained from the parallel computation which used 4 to 6 CPUs on the latest supercomputers.

The maximum capacity of VRAM (Video Random Access Memory) that can be used with a single GPU as of today is 6 GB on the Tesla C2070. It is the author' hope that the memory capacity of a single GPU will increase in the future and a large-scale simulation which involves tens of millions or more grid points will be made possible. On the other hand, the use of multi-GPUs systems is currently under consideration to perform largescale simulations which cannot be accommodated by the memory capacity of a single GPU. However, this approach requires additional internode parallelization using MPI (Message Passing Interface) or other means.

A significant reduction in computation time was also confirmed from parallel computation performed using a multi-cores CPUs (OpenMP) such as the Intel Core i7. Furthermore, a clear increase in computation speed was confirmed as a result of the Intel AVX instruction set, which has recently been made available for the latest CPUs (e.g., Intel Core i7 2600 K, 3.4 GHz, Sandy Bridge or Intel Core i9 7980XE Extreme Edition, 2.60 GHz, Skylake X). With the rapid improvement of hardware discussed above, it can be expected that unsteady turbulence analyses such as those by the RIAM-COMPACT will become the norm in the wind power industry.