Performance of a Two-Path Aliasing Free Calculation of a Spectral DNS Code

Mitsuo Yokokawa[1][0000−0003−3790−1243], Koji Morishita[2], Takashi Ishihara[3], Atsuya Uno[4], and Yukio Kaneda[5]

[1] Graduate School of System Informatics, Kobe University, Kobe 657-0013, Japan
yokokawa@port.kobe-u.ac.jp
[2] SOUM Corporation, Shibuya-ku, Tokyo 151-0072, Japan
[3] Graduate School of Environmental and Life Science, Okayama University, Okayama 700-8530, Japan
[4] RIKEN Center for Computational Science, Kobe 650-0047, Japan
[5] Center for General Education, Aichi Institute of Technology,Toyota 470-0392, Japan

Abstract. A direct numerical simulation (DNS) code was developed for solving incompressible homogeneous isotropic turbulence with high Reynolds numbers in a periodic box using the Fourier spectral method. The code was parallelized using the Message Passing Interface and OpenMP with two-directional domain decomposition and optimized on the K computer. High resolution DNSs with up to 12288^3 grid points were performed on the K computer using the code. Efficiencies of 3.84%, 3.14%, and 2.24% peak performance were obtainedi in double precision DNSs with 6144^3, 8192^3, and 12288^3 grid points, respectively. In addition, a two-path alias-free procedure is proposed and clarified its effectiveness for some number of parallel processes.

Keywords: Parallel computation, Fourier spectral method, Two-path de-aliasing method, DNS, K computer, Incompressible turbulence

1 Introduction

Recent rapid developments in the capacities and capabilities of supercomputers have led to computational approaches becoming powerful tools in turbulence studies. In particular, direct numerical simulations (DNSs) of turbulence can provide detailed turbulence data that are free from experimental uncertainties under well-controlled conditions.

In 2002, a highly efficient DNS code for incompressible turbulence based on the Fourier spectral method was developed for the Earth Simulator (ES), which was the fastest supercomputer at that time [1]. High-resolution DNSs of incompressible homogeneous isotropic turbulence with up to 4096^3 grid points and the Taylor scale Reynolds number $R_\lambda \sim 1200$ were performed on the ES using the code [2, 3]. We developed a new DNS code on the basis of the original spectral code used on the ES, and optimized it on the K computer[4] in 2013

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

to realize larger scale DNSs of turbulence with a higher Reynolds number (Re) than those reported so far. DNSs with up to 12288^3 grid points were carried out on the K computer and new results on the power law were obtained [5].

Though recent high-performance computer systems are characterized by a huge number of cores and a large amount of memory capacity, they are still insufficient for performing larger DNSs. Therefore, we have to consider an efficient algorithm for DNS code using the Fourier spectral method.

In this paper, the performance of the code evaluated on the K computer is described, and a parallel de-aliasing method, two-path aliasing free calculation, is also presented for the calculation of nonlinear terms of the ordinary differential equations (ODEs) discretized from the Navier-Stokes equations and continuum equation.

2 DNS Code Implementation

We consider homogeneous isotropic turbulence that obeys the incompressible Navier-Stokes equations and the continuum equation

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}, \quad \text{and} \tag{1}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

in a periodic box of side length 2π. Here $\boldsymbol{u} = (u_1, u_2, u_3)$, p, ν, and $\boldsymbol{f} = (f_1, f_2, f_3)$ denote velocity, pressure, kinematic viscosity, and external force, respectively (see [3] for details of the external force). Since boundary condition is periodic, the Fourier spectral method can be applied for discretization.

Then, equations (1) are written as

$$\left(\frac{d}{dt} + \nu|\boldsymbol{k}|^2\right)\hat{u_i}(\boldsymbol{k}) = \left(\delta_{ij} - \frac{k_i k_j}{|\boldsymbol{k}|^2}\right)\hat{h_j}(\boldsymbol{k}) + \hat{f_i}(\boldsymbol{k}), \tag{3}$$

where equation (2) is used to eliminate the pressure term. In equation (3), $\boldsymbol{k} = (k_1, k_2, k_3)$ is the wavenumber vector, $\hat{h_j}(\boldsymbol{k}) = ik_l\widehat{u_j u_l}$ is the nonlinear term, and the hat $\hat{}$ denotes the Fourier coefficients. The ODEs (3) are integrated with respect to time by the fourth-order Runge-Kutta-Gill method.

In calculating the nonlinear term $\hat{h_j}(\boldsymbol{k})$ which are expressed as convolution sums in the wave vector space, the fast Fourier transforms (FFT) can be efficiently used as the left normal series in Fig. 1(a). Aliasing errors generated by the spectral transform method are removed by the so-called phase shift method (Fig. 1) and the cut-off beyond the maximum wavenumber $k_{\max} \equiv (\sqrt{2}/3)N$, where N is the number of grid points in each of the Cartesian coordinates in the physical space.

2.1 Code Parallelization

In DNS using the spectral method, most of the computation time is associated with evaluating the alias-free nonlinear terms, and the three-dimensional FFTs

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

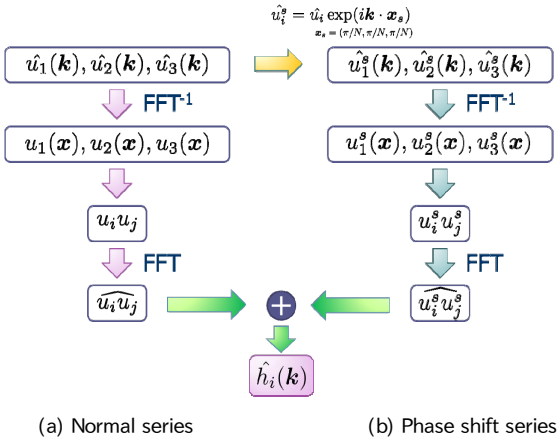

Fig. 1. Phase shift method

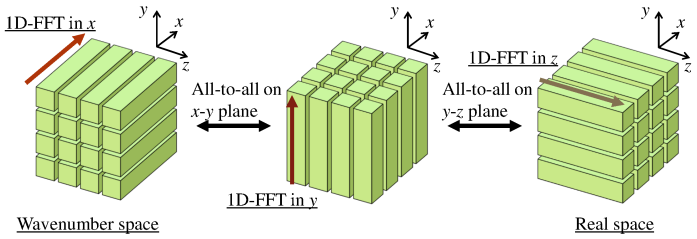

Fig. 2. Schematic diagram of the 3D-FFT procedure used in the base code.

(3D-FFTs) are used in the evaluation, which requires data transfer among all parallel processes in the parallel computation.

We have developed a parallel DNS code (referred to as **base code**) in Fortran 90 using the Message Passing Interface (MPI) and OpenMP for parallel processing, and we optimized this code for the K computer. Two-directional domain decomposition or pencil decomposition for MPI parallelization is applied for the data distribution and the FFTW library [6] is used in the implementation.

Figure 2 shows a parallel 3D-FFT procedure used in the code. The complex valued data in the wavenumber space are partitioned among MPI processes in y- and z-directions. After executing the one-dimensional FFTs (1D-FFTs) in the x-direction, the domain decomposition changes to the z- and x-directions by performing all-to-all communications for the x-y plane processes. After executing the 1D-FFTs in the y-direction, the domain decomposition changes to the x- and y-directions by performing all-to-all communications for the y-z plane processes in the same manner. Finally, the 1D-FFTs are executed in the z-direction. We created sub-communicators, each of which contains several x-y or

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

y-z planes, using the MPI function `MPI_COMM_SPLIT`, so that all-to-all communication can be confined in each two-dimensional plane (or slab) contained in each sub-communicator.

2.2 Two-Path Calculation of Nonlinear Terms

The evaluation of nonlinear terms represents a time-consuming part in the code. Considering that two paths of normal and phase shift series of transform methods can be calculated concurrently as shown in Fig. 3, the calculation time can be shortened by using twice number of parallel processes, if available.

One set of parallel processes computes the nonlinear terms of the normal series and the other set of parallel processes computes those of the phase shift series. In this case, the number of parallel processes are equal in both sets. Computation parts of the base code other than the computation of nonlinear terms is executed redundantly in both sets to avoid the additional data transfer in the wavenumber space. The computation results of nonlinear terms in each series should be exchanged between both sets to calculate the sum of them, and therefore additional send/receive operations are incorporated in this implementation. Thus computation time of nonlinear terms is expected to decrease to half. We also implemented and evaluated this revised phase shift method in the code (referred to as **revised code**).

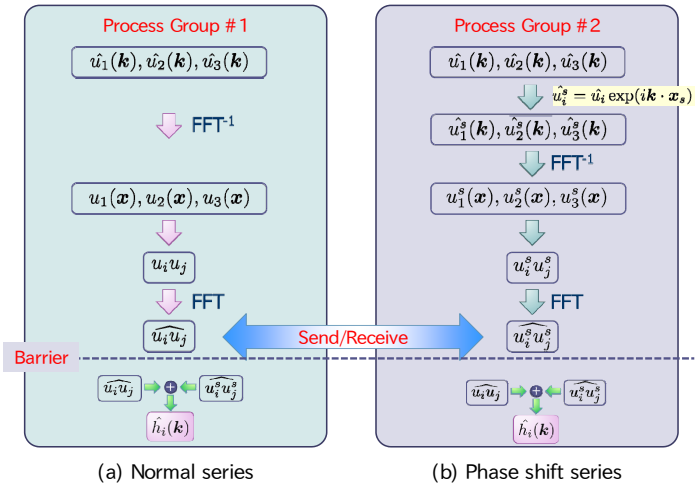

Fig. 3. Two-path phase shift method

3 Simulation Results and Code Performance

We performed DNSs of incompressible turbulence in the periodic boundary box with grid points $N^3 = 4096^3$, 6144^3, 8192^3, and 12288^3 on the K computer. In

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

this section, brief results were followed by performance evaluations of the newly developed code. An extensive and detailed analysis of the DNSs results from the perspective of turbulence were reported in [5].

3.1 Simulation results

Calculated velocities in the box were used to analyze the vortex dynamics in high Reynolds (Re) turbulent flows. Figure 4 depicts an iso-surface of large vorticity area $|\omega| = 6\langle\omega^2\rangle^{1/2}$ in the DNS with grid points $N^3 = 12288^3$ at an approximately statistical steady state. Similar figures are seen in Ref. [7]. This figure was drawn by an originally developed visualization code based on the marching cubes algorithm [8]. Visualization data can be generated in part with a few layers of grid points in the box and a visualization processing of a batch of layers can be applied independently to the other batch. Therefore the visualization can be carried out in parallel and on computational resources which are not much in performance.

The figure shows that there are tube-like vortex-clusters at various sizes from some integral length scale L down to several dozen Taylor micro scale λ. The lengths L and λ are shown in Fig. 4. It also shows that distinct vortex-cluster areas and void (vorticity-free) areas coexist in high Re turbulence.

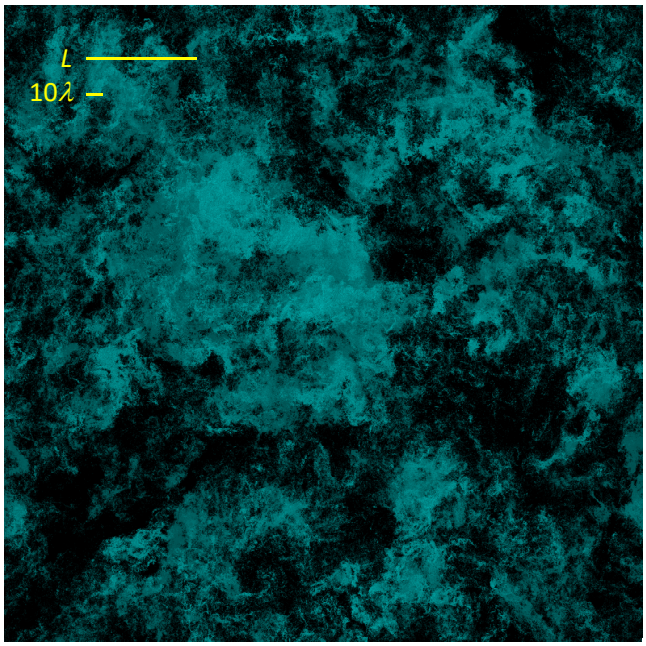

Fig. 4. Isosurface of vorticity $|\omega| = 6\langle\omega^2\rangle^{1/2}$.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

Table 1. Sustained performance of the base code on the K computer.

N^3	Number of compute nodes	Performance (Tflop/s)	Efficiency (%)
4096^3	2,048 (64 × 32)	10.80	4.12
	4,096 (64 × 64)	12.48	2.38
6144^3	6,144 (96 × 64)	30.16	3.84
	12,288 (96 × 128)	44.03	2.80
8192^3	8,192 (128 × 64)	32.76	3.14
	16,384 (128 × 128)	42.01	2.00
12288^3	24,576 (192 × 128)	70.52	2.24
	49,152 (384 × 128)	128.45	2.04

3.2 Computational Performance

We measured the sustained performance of the base code using the hardware counter on the K computer. An MPI process is assigned to each compute node, and therefore the number of MPI processes equals to the number of compute nodes.

Table 1 shows the sustained performance of the code calculated in double precision with $N^3 = 4096^3$, 6144^3, 8192^3, and 12288^3 grid points. Here, note that the interconnect network of the K computer has a Tofu basic unit consisting of 12 compute nodes and 1D-FFT for grid points with a multiple of 3 is efficiently carried out. Thus, we took 6144^3 and 12288^3 grid points which are multiples of 3 into consideration.

We observe that performance (Tflop/s) increases with the number of compute nodes, and the highest performance is 128.45 Tflop/s for $N^3 = 12288^3$ with 49,152 compute nodes. The maximum efficiency is 4.12%, which was achieved for $N^3 = 4096^3$ using the smallest number of compute nodes (2,048 compute nodes). Higher efficiency is obtained with fewer compute nodes for the same value of N^3, because the ratio of communication time to calculation time increases as the number of compute nodes increases.

3.3 Computation time of the revised code

Computation time of the base and revised codes were measured for the DNS with 768^3 grid points and 100 time steps. Three cases were considered, as described below.

Case 1 Computation time of the base code was measured with 96 MPI processes on 96 compute nodes of the K computer.

Case 2 Computation time of the revised code with 192 MPI processes on 192 compute nodes. These MPI processes were divided into two sets and the computation of normal series and shifted series is assigned to each set. The message size of each all-to-all communication in 3D-FFT is the same as that in case 1 but additional send/receive operation for the exchange of nonlinear term values occurs between the two sets.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

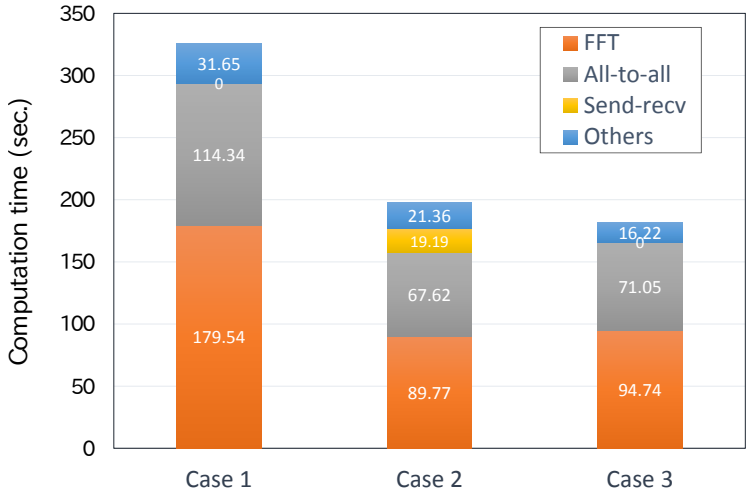

Fig. 5. Computation time of the three cases

Case3 Computation time of the base code with 192 MPI processes. The message size of each all-to-all communication is halved but the number of counter nodes to be communicated in all-to-all communications is doubled.

Figure 5 shows the computation time for all cases. The computation time associated with case 2 is approximately half that of case 1 and additional send/receive time (approximately 10% of the total time) is observed. The revised phase shift method appears to work well. Further, computation time associated with case 3 (i.e. base code with double amount of MPI processes compared to case 1) is also half of that of case 1 and less than that of case 2. No time is spent on send/receive operations in case 3. To some extent the base code exhibits strong scaling as the number of MPI processes increases, and therefore the computation time of case 3 is half of that associated with case 1.

If a sufficient number of compute nodes can be used, and if an appropriate number of DNS parallelisms corresponding to the number of compute nodes for a given number of grid points can be found, the base code exhibits better performance than the revised code. However, where there are a large number of compute nodes available and no more parallelisms can be extracted from a DNS for a given number of grid points, which is substantially less than the number of compute nodes, the revised phase shift method appears to be useful. For example, DNS with 128^3 grid points normally has up to 128^2 parallelisms where pencil decomposition is carried out on a parallel system with greater than 256^2 nodes, and the base code cannot be assigned properly in this case but the revised two-path phase shift method can work efficiently.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

4 Conclusions

To realize DNSs of turbulence with a high Reynolds number in a periodic box, we developed a parallel DNS code for the K computer. Two-directional domain decomposition or pencil decomposition for the data distribution in process parallelization by MPI is applied. By using sub-communicators in MPI, we realized an efficient communication procedure that does not involve all-to-all process communication among all processes in the parallel processing of the 3D-FFT.

The code developed in the study is applicable to non-Newtonian fluid, in the sense that it is applicable to the nonlinear convection term whose form is the same as the one for Newtonian fluid. Moreover, parallelization techniques by domain decomposition used in the code are applicable to the computation concerning with the wall-parallel directions of turbulent channel flow, and also to the wall-normal direction if one takes the fast Fourier-cosine transforms in the direction.

Efficiencies of 3.84%, 3.14%, and 2.24% peak performance were obtained in double precision calculations of DNSs with 6144^3, 8192^3, and 12288^3 grid points, respectively, on the K computer. Using the new code, we performed DNSs of turbulence on a larger scale than hitherto reported in the literature, specifically, with respect to the Reynolds number. This led to attaining a higher Reynolds number ($R_\lambda \sim 2300$, $Re \sim 1.5 \times 10^5$) than has thus far been reported at an approximately statistical steady state (see [5]).

A two-path de-aliasing calculation technique of the phase shift method for the alias-free calculation of nonlinear terms was also proposed, and its efficiency was explored for the case where there are many compute nodes relative to the number of DNS parallelisms with less grid points than parallel processes. Next-generation supercomputers will have many more cores compared to those of contemporary high-performance systems, thus computation time can be reduced by two-path de-aliasing technique.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number (C) 26390130 and obtained by using computational resources of the K computer provided by the RIKEN Advanced Institute for Computational Science through the HPCI System Research projects (Project IDs: hp140135, hp150174, hp160102, hp170087, hp180109, and hp190076).

References

1. Yokokawa, M., Itakura, K., Uno, A., Ishihara, T., Kaneda, Y.: 16.4-Tflops Direct Numerical Simulation of Turbulence by a Fourier Spectral Method on the Earth Simulator, Proc. of ACM/IEEE Conf. on Supercomputing, pp. 1–17, IEEE, Baltimore, U.S.A. (2002). https://doi.org/10.1109/SC.2002.10052

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44

2. Kaneda, Y., Ishihara, T., Yokokawa, M., Itakura, K., Uno, A.: Energy Dissipation Rate and Energy Spectrum in High Resolution Direct Numerical Simulations of Turbulence in a Periodic Box, Phys. Fluids, **15**(2), pp. L21–L24 (2003). https://doi.org/10.1063/1.1539855
3. Ishihara, T., Kaneda, Y., Yokokawa, M., Itakura, K., Uno, A.: Small-scale statistics in high-resolution direct numerical simulation of turbulence: Reynolds number dependence of one-point velocity gradient statistics, J. Fluid Mech., **592**, pp. 335–366 (2007). https://doi.org/10.1017/S0022112007008531
4. Yokokawa, M., Shoji, F., Uno, A., Kurokawa, M., Watanabe, T.: The K computer: Japanese next-generation supercomputer development project, IEEE/ACM International Symposium on Low Power Electronics and Design, pp. 371–372, IEEE, Fukuoka, Japan (2011). https://doi.org/10.1109/ISLPED.2011.5993668
5. Ishihara, T., Morishita, M., Yokokawa, M., Uno, A., Kaneda Y.: Energy spectrum in high-resolution direct numerical simulations of turbulence, Phys. Rev. Fluids, **1**(8), 082403 (2016). https://doi.org/10.1103/PhysRevFluids.1.082403
6. Frigo, M., Johnson, S. G.: The Design and Implementation of FFTW3, Proc. of the IEEE, **93**(2), 216–231 (2005). https://doi.org/10.1109/JPROC.2004.840301
7. Ishihara, T., Uno, A., Morishita, M., Yokokawa, M., Kaneda Y.: Vortex Clusters and Their Time Evolution in High-Reynolds-Number Turbulence, NAGARE, **35**(2), pp. 109–113 (2016). (*in Japanese*)
8. Lorensen, W.E., Cline, H.E.: Marching Cubues: A high resolution 3D surface construction algorithm, Computer Graphics, **21**(4), 163–169 (1987). https://doi.org/10.1145/37402.37422

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22747-0_44