Development of numerical techniques toward extreme scale fusion plasma turbulence simulations

Yasuhiro Idomura[1], Susumu Yamada[1], Toshiyuki Imamura[2], Motoki Nakata[1], and Masahiko Machida[1]

[1]Japan Atomic Energy Agency, Japan
[2]The University of Electro-communications, Japan

1 Introduction

The magnetic confinement fusion is one of the most promising approaches to a fusion reactor, and in the next generation experiment, the International Thermonuclear Experimental Reactor (ITER), the scientific and technological feasibility of a magnetic fusion reactor will be demonstrated. Fusion plasmas are extremely complex physical system consisting of multiple fluids (electrons and multiple species ions) coupled through electromagnetic fields and weak Coulomb collisions. Because of this complexity, computer simulations have been established as essential tools in wide spectrum of fusion science. Among several critical issues, the turbulent transport is the most demanding issue, because five dimensional (5D) gyrokinetic simulations [1, 2] are needed for studying turbulent fusion plasmas. So far, we have been developing a global Gyrokinetic Toroidal 5D Eulerian code (GT5D) [3]. GT5D is based on the so-called global full-f approach, in which the equilibrium part f_0 and turbulent perturbation δf of particle distribution are solved simultaneously using the same first principles in a full torus computational domain. In contrast to conventional δf simulations, which solve only δf with fixed f_0 by assuming complete spatio-temporal scale separation between f_0 and δf, full-f simulations can simulate relevant multi-scale phenomena such as interactions between macro-scale mean flows and micro-scale turbulence and evolutions of temperature profiles affected by turbulent energy transport in a self-consistent manner. The full-f approach requires robust and accurate numerical treatments, which can resolve small amplitude perturbations ($\delta f/f_0 \sim 1\%$) for time scales much longer than a turbulent correlation time. Such numerical requirements were satisfied by developing a Non-Dissipative Conservative Finite Difference (NDCFD) scheme [4]. The accuracy of GT5D was demonstrated through quantitative verification studies [3]. In addition, validation studies [5, 6] captured qualitative transport properties in the experiment, such as the stiffness of temperature profiles, intermittency of avalanche-like heat transport, spontaneous plasma rotation, and the plasma size scaling of the heat diffusivity. Although the rapid progress of computing power enhanced capabilities of GT5D, we need to establish the path towards next generation Peta-scale and Exa-scale computing for simulating the ITER, which is several times larger than existing fusion devices. In such extreme scale computing, severe requirements on the memory usage and the parallel efficiency are expected. In this work, we present numerical techniques required for extreme scale fusion plasma turbulence simulations.

2 Calculation models

We consider the electrostatic ion turbulence described by gyrokinetic ions and adiabatic electrons in a tokamak configuration, which consists of nested magnetic surfaces with an axisymmetric torus topology. GT5D is based on the modern gyrokinetic theory [1], which describes dynamics of the plasma particle distribution f in 5D phase space $\mathbf{Z} = (t; \mathbf{R}, v_\parallel, \mu)$. Here, $\mathbf{R}$ denotes the configuration space, $\mathbf{v}$ is the velocity of particles, $v_\parallel = \mathbf{b} \cdot \mathbf{v}$ and $v_\perp = |\mathbf{b} \times \mathbf{v}|$ are the velocities in the parallel and perpendicular direction to the magnetic field, $\mu = m_i v_\perp^2 / 2B$ is the magnetic moment, m_i is the mass of ions, and B is the magnetic field. A conservative form of the gyrokinetic equation is given as

$$\frac{\partial \mathcal{J} f}{\partial t} + \nabla \cdot (\mathcal{J} \dot{\mathbf{R}} f) + \frac{\partial}{\partial v_\parallel} (\mathcal{J} \dot{v}_\parallel f) = \mathcal{J} C(f) + \mathcal{J} S_{src}, \quad (1)$$

where $\mathcal{J}$ is the Jacobian of coordinates, and $\dot{\mathbf{R}}$ and $\dot{v}_\parallel$ are Hamiltonian flows, which are calculated using the magnetic field B and the turbulent electrostatic potential ϕ, and satisfy an incompressible condition, $\nabla \cdot (\mathcal{J} \dot{\mathbf{R}}) + \partial_{v_\parallel} (\mathcal{J} \dot{v}_\parallel) = 0$, by their definition. The rhs shows a source term S_{src} and a Coulomb collision term $C(f)$, which is given by a linear convection-diffusion operator based on the linear Fokker-Planck collision model [7]. The turbulent electrostatic potential ϕ is solved by using the gyrokinetic Poisson equation,

$$-\nabla_\perp \cdot P_1 \nabla_\perp \phi + P_2 \left(\phi - \langle \phi \rangle_f \right) = g - n_0, \quad (2)$$

where P_1, P_2 are coefficients given by plasma parameters, g is the ion density computed from f, n_0 is the equilibrium electron density, and $\langle \cdot \rangle_f$ is an averaging operator over the magnetic surface.

GT5D consists mainly of three solvers, the Gyrokinetic solver (4D convection operator in the lhs of Eq.(1)), the Collision operator (2D convection-diffusion operator in the rhs of Eq.(1)), and the Field solver (Eq.(2)). The conservative gyrokinetic equation (1) is discretized using the 4th order accurate NDCFD, while the collision operator uses the 6th order accurate centered finite difference. In the time-integration, the Courant-Friedrichs-Lewy (CFL) condition is determined by the linear part of the 4D convection operator, which involves the thermal motion of particles in the direction of magnetic field. Therefore, we adopt the 2nd order accurate additive semi-implicit Runge-Kutta method (ASIRK) [8], and treat the stiff linear term in an implicit manner. The gyrokinetic Poisson equation (2) is solved using a Fourier mode expansion in the azimuthal angle of torus and a 2D finite element approximation on the cross section of torus.

3 Numerical techniques

In order to improve the parallel efficiency, the following key techniques are developed. The first one is a multi-layer hybrid parallelization model [9], in which multi-dimensional domain decomposition is designed following the physical symmetry properties of each solver, and all the communications within and among three solvers are implemented on a single layer of the hierarchical network consisting of multiple MPI layers and a SMP layer. This implementation avoids collective communications among all the MPI processes, and suppresses the size of each MPI communicator below ~ 100 up to $\sim 10^6$ MPI processes. In addition to this global network design, we apply novel communication overlap techniques to each communications. The communication overlap techniques use either non-blocking communications with additional control sequences or communication threads in a MPI/OpenMP hybrid parallel model. Our communication overlap techniques are based on conventional MPI libraries, and work on most of existing platforms based on dedicated and commodity networks.

On the other hand, the following low memory usage techniques are developed. Firstly, in a flat-MPI implementation, the sizes of buffers used in domain decomposition and in MPI processes show an explosive growth beyond $\sim 10^4$ processes. To avoid this issue, a MPI/OpenMP hybrid parallel model is adopted, and the number of MPI processes is reduced by an order of magnitude. Secondly, the size of the 2D finite element matrix in the Field solver exceeds $\sim$ 100GB for ITER-size parameters. A filtering technique [10] is developed based on the physical property of fusion plasma turbulence, which is aligned to the magnetic field reflecting a resonance condition. By applying the filtering technique, the size of finite element matrix is reduced by three orders of magnitude, and the Field solver is implemented using a direct solver in LAPACK. Finally, the implicit part of ASIRK is given by a huge asymmetric matrix with 17 stencils (5 points in each direction of 4D grids). Since the problem size is typically $\sim 10^{10}$, it is difficult to store such a huge matrix even with sparse matrix storage formats, and therefore, general iterative matrix solver libraries are not available. To overcome this difficulty, massively parallel Krylov subspace solvers with optimized domain decomposition are developed by implementing the matrix as finite difference equations. The original solver was based on a generalized conjugate residual (GCR) method [11]. In recent work [12], it is extended by adopting preconditioning techniques and advanced iterative solver algorithms such as GMRES [11] and BiCGstab [11]. By using extended solvers, both the number of iterations and the computational cost are reduced, and the processing efficiency on scalar processors with relatively low memory bandwidth is improved.

4 Strong scaling of GT5D on K

Table 1 shows a summary of performances observed in strong scaling of GT5D on K. In this problem size, the size of finite element matrix becomes $\sim$ 6.5 GB, while it is reduced to $\sim$ 74 MB with the filtering technique. At 2,048 cores (256 nodes), the total memory usage is $\sim$ 1.8 TB. The memory usage is significantly increased with the number of MPI processes, and at 32,768 cores, it is increased by $\sim$ 2.77 times. Since the memory usage is determined almost by the number of MPI processes, this data indicates that the memory usage is significantly reduced by the MPI/OpenMP hybrid parallel model, where the number of MPI processes is reduced by 1/8 on K. The peak ratio of sustained performance with (without) the communication overlap techniques is varied from $\sim$ 13.1% ($\sim$ 10.8%) at 2,048 cores to $\sim$ 10.4% ($\sim$ 9.1%) at 32,768 cores. Thanks to the multi-layer hybrid parallelization model, GT5D shows reasonably good strong scaling even without the communication overlap techniques. However, by applying the communication overlap techniques, the ratio of communication cost is dramatically reduced and is suppressed below $\sim$ 10% up to 32,768 cores.

Table 1. The memory usage, the sustained performance, and the ratio of communication cost observed in strong scaling of GT5D on K with 8 SMP threads. The problem size is $240 \times 64 \times 240 \times 128 \times 32$ for 5D grids and 192×384 for 2D finite elements. ( ) shows the data without communication overlap techniques.

Cores	Mem.(TB)	Perf.(TF)	Comm.(%)
2,048	1.8(2.2)	4.3 (3.6)	2.9(14.0)
4,096	2.0(2.4)	8.4 (6.3)	4.1(13.6)
8,192	2.5(2.9)	16.4(13.5)	5.6(18.3)
16,384	3.2(3.6)	29.7(24.8)	5.4(17.8)
32,768	5.0(5.4)	54.5(47.6)	8.9(23.4)

5 Conclusion

Key requirements in next-generation Peta-scale or Exa-scale fusion plasma turbulence simulations are lower memory usage and extreme parallelism. To achieve these requirements, novel numerical techniques are developed on GT5D in various levels ranging from physics models to computational models. The performances of these techniques show promising features on K.

Acknowledgement

This work is supported by the MEXT grant for HPCI Strategic Program Field No.4: Next-Generation Industrial Innovations. A part of the results is obtained by early access to the K computer at the RIKEN-AICS. One of the authors (Y.I.) is supported also by the MEXT (Grant No. 22866086) and by the G8 Research Councils Initiatives (the Nuclear Fusion Simulations @ Exa-scale project).

References

[1] X. Garbet, *et al.*, Nucl. Fusion **50**, 043002(2010).

[2] Y. Idomura, *et al.*, C. R. Physique **7**, 650 (2006).

[3] Y. Idomura, *et al.*, Comput. Phys. Commun. **179**, 391 (2008).

[4] Y. Idomura, *et al.*, J. Comput. Phys. **226**, 244 (2007).

[5] Y. Idomura, *et al.*, Nucl. Fusion **49**, 065029 (2009).

[6] S. Jolliet, *et al.*, Nucl. Fusion **52**, 023026 (2012).

[7] X. Q. Xu, *et al.*, Phys. Fluids B **3**, 627 (1991).

[8] X. Zhong, J. Comput. Phys. **128**, 19 (1996).

[9] Y. Idomura and S. Jolliet, in Proceedings of SC11 (2011).

[10] S. Jolliet, *et al.*, J. Comput. Phys. **231**, 745 (2012).

[11] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition*, SIAM (2003).

[12] S. Yamada, *et al.*, in this conference.