

## ARTICLE

## Simulating Plasma Turbulence with the Global Eulerian Gyrokinetic Code GT5D

Sébastien JOLLIET\* and Yasuhiro IDOMURA

Japan Atomic Energy Agency, Higashi Ueno 6-9-3, Taitou, 110-0015 Tokyo, Japan

The efficiency of future fusion devices such as ITER is strongly affected by plasma turbulence, which produces outwards particle, momentum and heat transport. The most advanced tools to study such problems are gyrokinetic codes, which solve the Boltzmann-Maxwell system in five-dimensional phase-space. In this work, the gyrokinetic global Eulerian code GT5D is presented, focusing on the different numerical schemes and on the parallelization. Weak and strong scaling shows excellent scalability up to  $10^4$  processors. Finally, a straight-field-line solver is presented, which, for large plasma size, reduces the total memory by two orders of magnitude. Using Fourier transforms, this solver takes advantage of the alignment of turbulence with the magnetic field lines to reduce 3D turbulent fields to quasi-2D by filtering out non-aligned components. This procedure does not affect the steady state of GT5D simulations. Then, the code is applied to study the influence of plasma size scaling on plasma turbulence: worse-than-Bohm scaling is found for the first time in experimentally realistic gyrokinetic simulations.

**KEYWORDS:** *plasma turbulence, gyrokinetics, ITG*

## I. Introduction

In fusion devices such as Tokamaks, the measured particle, momentum and heat transport perpendicular to the magnetic field greatly exceeds the neoclassical (resulting from Coulomb collisions) predictions. This so-called *anomalous transport* is attributed to plasma turbulence: gradients of density and temperature from the core to the edge of the plasma provide a source of free energy for electromagnetic perturbations to develop. The radially outwards transport degrades the energy confinement time  $\tau_E$ , hence the fusion performance. Numerical simulations of plasma turbulence are extremely complex and require state-of-the-art computers: firstly, although the gyrokinetic theory reduces the dimensionality from 6 to 5 by getting rid of the fast (compared to the turbulent time scale) gyromotion of particles, the number of grid points is typically of the order of  $10^9$ . Time steps  $\Delta t \sim 10^{-7} s$  must be used but profiles typically evolve on a collision time  $\tau_{ii} \sim 10^{-3} s$ . Global gyrokinetic simulations typically require  $10^5$ - $10^6$  CPU hours. Realistic gyrokinetic simulations will need to solve simultaneously the ion and the electron dynamics, increasing the total cost by  $\sqrt{(m_i/m_e)} \sim 60$  for Deuterium plasmas. One of the main parameters of turbulence is  $\rho^* = \sqrt{(\tau)} \rho_L / a$ , where  $\tau = T_e / T_i$  is the ion to electron temperature ratio,  $\rho_L$  is the Larmor radius and  $a$  is the minor radius of the Tokamak.  $(\rho^*)^{-1}$  defines the spatial resolution, and will increase by a factor 2-3 in future fusion devices such as ITER, for which  $(\rho^*)^{-1} \sim 10^3$ . Global gyrokinetic simulations of ITER retaining all relevant physical effects are unreachable with today's supercomputers. Consequently, many efforts must be undertaken to optimize the numerical algorithms and parallelization of gyrokinetic codes.

This paper presents the gyrokinetic global Eulerian GT5D code,<sup>1)</sup> a state-of-the-art code to study Ion Temperature Gradient (ITG) turbulence. GT5D memory scales like  $(\rho^*)^{-3}$  and rapidly becomes a burden. This paper presents the implementation of a new straight-field-line solver<sup>2)</sup> which greatly reduces the required memory and will enable the simulation of much larger plasmas.

The rest of the paper is organized as follows. The gyrokinetic equations and the associated numerical schemes are presented in Section II. The parallelization and scaling properties are detailed in Section III. In Section IV, the implementation of the straight-field-line solver is presented and some simulation results are shown in Section V. Finally, conclusions are given in Section VI.

## II. The GT5D Code

## 1. Gyrokinetic Model

A full description of GT5D can be found in Reference 3. The main numerical aspects are presented here for completeness. The Boltzmann equation reads:

$$\frac{\partial J_{ps} f}{\partial t} + \nabla \cdot (J_{ps} \mathbf{R} f) + \frac{\partial}{\partial v_{\parallel}} (J_{ps} \dot{v}_{\parallel} f) = J_{ps} [C(f) + S_{src}(f) + S_{snk}(f)]. \quad (1)$$

Where  $f(R, v_{\parallel}, \mu, t)$  is the ion guiding distribution function,  $\mathbf{R}$  is the guiding-center position,  $v_{\parallel}$  is the velocity parallel to the magnetic field  $\mathbf{B}$ ,  $\mu$  is the magnetic moment and  $J_{ps}$  is the phase space Jacobian. The equations of motions are obtained through:

\*Corresponding author, E-mail: jolliet.sebastien@jaea.go.jp

$$\dot{A} \equiv \{A, H\} = \frac{\Omega_i}{B} \left( \frac{\partial A}{\partial \alpha} \frac{\partial H}{\partial \mu} - \frac{\partial A}{\partial \alpha} \frac{\partial H}{\partial \mu} \right) + \frac{\mathbf{B}^*}{m_i B_{\parallel}^*} \cdot \left( \nabla A \frac{\partial H}{\partial v_{\parallel}} - \nabla H \frac{\partial A}{\partial v_{\parallel}} \right) - \frac{c}{q_i B_{\parallel}^*} \mathbf{b} \cdot (\nabla A \times \nabla H) \quad (2)$$

$$H = \frac{1}{2} m_i v_{\parallel}^2 + \mu B + q_i \langle \phi \rangle_{\alpha}. \quad (3)$$

Where  $\mathbf{B} = B\mathbf{b}$ ,  $\mathbf{B}^* = \mathbf{B} + Bv_{\parallel}/\Omega_i \nabla \times \mathbf{b}$ ,  $B_{\parallel}^* = \mathbf{b} \cdot \mathbf{B}^*$ ,  $\Omega_i = q_i B / (m_i c)$  is the cyclotron frequency and  $\langle \cdot \rangle_{\alpha} = 1/(2\pi) \int d\alpha$  is the gyro-averaging operator, where  $\alpha$  is the gyophase angle. The magnetic field can either be specified analytically or obtained numerically from an equilibrium solver. It is axisymmetric and can be written  $\mathbf{B} = B_{\phi} \mathbf{e}_{\phi} + B_{\theta} \mathbf{e}_{\theta}$ , where  $B_{\phi}$  is the toroidal component and  $B_{\theta}$  is the poloidal component.

The collision operator  $C(f)$  is a linearized, drift-kinetic Fokker-Planck operator<sup>4)</sup>  $C(f) = C_T(f) + C_F(f)$  where  $C_T(f)$  is the test-particle operator and  $C_F(f)$  is the field-particle operator. In particular the field-particle operator is constructed numerically in order to conserve density, parallel momentum and energy up to machine precision.<sup>5)</sup>

The source operator is  $S_{src} = A_{src}(\mathbf{R}) \tau_{src}^{-1} (f_{M1} - f_{M2})$ , where  $A_{src}$  is a deposition profile,  $f_{M1}$  and  $f_{M2}$  are (shifted) Maxwellian distributions and  $\tau_{src}$  is a time constant.  $\tau_{src}$  is set by imposing no particle and momentum input but fixed power input  $P_{in}$ .

$$0 = \int S_{src} d^6 Z = \int m_i v_{\parallel} S_{src} d^6 Z$$

$$P_{in} = \int \left( \frac{1}{2} m_i v_{\parallel}^2 + \mu B \right) S_{src} d^6 Z \quad (4)$$

The sink operator is  $S_{snk} = A_{snk}(\mathbf{R}) \tau_{snk}^{-1} (f_0 - f)$ , where  $A_{snk}$  is a deposition profile,  $f_0$  is the initial distribution and  $\tau_{snk}$  is a time constant.

Self-consistency is imposed by the quasi-neutrality equation:

$$-\nabla_{\perp} \cdot \frac{\rho_{ii}^2}{\lambda_{Di}^2} \nabla_{\perp} \phi + \frac{1}{\lambda_{De}^2} (\phi - \langle \phi \rangle_f) = 4\pi e \left[ \int f \delta(\mathbf{R} + \rho - \mathbf{x}) d^6 Z - n_{0e} \right]. \quad (5)$$

Where  $\mathbf{R} + \rho$  is the particle position,  $d^6 Z = \int_{ps} d\mathbf{R} dv_{\parallel} d\mu d\alpha$  is the phase space volume,  $\rho_{ii}$  is the Larmor radius evaluated with the thermal velocity,  $\lambda_{Di}, \lambda_{De}$  are the ion and electron Debye lengths,  $\langle \cdot \rangle_f$  is a flux-surface-average operator and  $n_{0e}$  is the equilibrium electron density. Electrons are adiabatic. Dirichlet boundary conditions are imposed at the plasma edge, while a free boundary is imposed at the magnetic axis.

## 2. Implementation

The kernel of the code aims at solving Eq. (1) with an Eulerian scheme. The distribution function is therefore discretized on a 5D grid  $N_R, N_Z, N_{\phi}, N_{v_{\parallel}}, N_{v_{\perp}}$ , where  $(R, \phi, Z)$  are cylindrical coordinates, and is evolved using a Non-Dissipative Conservative Finite Difference scheme (NDCFD).<sup>3)</sup> This scheme ensures the numerical conservation of the L1 and L2 norms, which are important for the stability

and accuracy of a conservative Vlasov simulation. The advection of the distribution function is split into the linear stiff motion and the nonlinear motion. The latter is advected explicitly while the linear motion is solved implicitly, based on a additive semi-implicit Runge-Kutta method (ASIRK)<sup>6)</sup> in order to decrease the Courant-Friedrich-Lewy number. A huge linear implicit operator ( $\sim 10^9$  degrees of freedom) is solved using a generalized conjugate residual method, which converges normally with several tens of iterations. Another advantage of separating the motion in this way is that both the linear and nonlinear Hamiltonian functions are 4D functions, which greatly reduces the needed memory.

Velocity derivatives of the distribution function appearing in the collision operator are computed with a 6th order centered finite difference scheme.  $C_T(f)$  is calculated at each spatial grid point, and  $C_F(f)$  is computed numerically to enforce the conservation of density, momentum and energy. Therefore, at each time step a 3x3 matrix is solved at each spatial grid point.

Then, the Poisson equation is solved using quadratic B-splines finite elements on a 2D grid on flux coordinates. The toroidal angle is treated in Fourier space. This scheme transforms Eq. (5) into a linear system. The matrix is hermitian, positive-definite with  $3(N_r + 2)N_z^2$  complex elements, where  $N_r$  and  $N_z$  are the finite elements grid resolution. The number of unknowns is typically  $10^7 - 10^8$ . The flux-surface-average operator must be treated separately: 2 matrices must be stored. The linear system is then solved with a direct LAPACK solver using a LU decomposition.

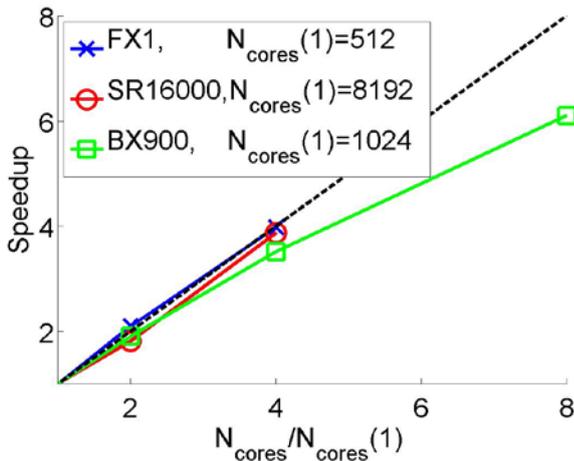
## III. Parallelization

### 1. Description

Evolving a distribution function of more than  $10^9$  grid points is of course impossible without parallelization. GT5D uses hybrid MPI and Open-MP parallelization. The MPI-domain is a 3D domain  $(\mu, R, Z)$ . Since  $\mu$  is a constant of motion, it only appears parametrically in the equations. Communications in the  $\mu$  direction must be performed when one computes  $\mu$  integrals (r.h.s. of Eq. (5)) or derivatives (when computing  $C_T(f)$ ). In the  $(R, Z)$  direction, the value of boundary cells must be communicated after each advection. The quasi-neutrality Eq. (5) is parallelized in the toroidal angle direction  $\phi$  ( $n$  direction in Fourier space) using the  $(R, Z)$  communicator. All-to-all communications must be performed to transpose the data from the  $(R, Z)$  to the  $n$  parallelization. Such large parallel data transpose are also needed for the collision operator, since the  $\mu$  values of the distribution function are stored on different processors.

### 2. Scalability

**Figure 1** shows the strong scaling speedup of GT5D on different machine architectures, for a problem size of  $240 \times 240 \times 64 \times 128 \times 32$ . It is excellent up to 32,768 cores on the SR16000 machine and up to 2,048 cores on the FX1 machine. The scaling is good up to 8,192 processors on the BX900 machine. Note that on this architecture, the peak performance is 11.5% at 8,192 cores and goes up to 13.2% for typical production runs at 4,096 processors. The peak



**Fig. 1** Strong scaling speedup achieved with GT5D on a Fujitsu FX1 machine (blue, crosses) from 512 to 2,048 cores, on a Hitachi SR16000 machine (red, circles) from 8,192 to 32,768 cores and on a Fujitsu BX900 (green squares) from 1,024 to 8,192 cores. The dashed line is the ideal scaling.

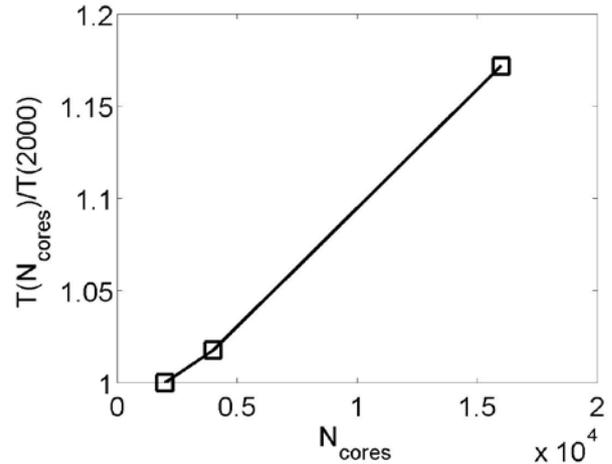
performance is 10.5% on the FX1 machine at 2,048 cores and 8.7% on the SR16000 at 32,768 cores.

The code performance can be improved by using OpenMP parallelism on the BX900 machine. At fixed problem size and number of cores, using OpenMP up to 4 threads decreases the cost MPI communications and increases the sustained performance. **Figure 2** shows a weak scaling of GT5D for a fixed plasma size  $\rho^*=1/600$  (the largest that can enter the BX900 machine). The number of grid points per process is kept fixed ( $4 \cdot 10^6$ ) and the number of points in the toroidal direction is scaled proportionally with the number of processors (this procedure is physically relevant as one does not need to simulate a full torus in the toroidal direction to obtain a converged value of heat transport<sup>7</sup>). The largest resolution is  $640 \times 640 \times 256 \times 80 \times 20$ . By going from 2,000 to 16,000 cores (4 open MP PEs, 500 to 4,000 MPI PEs), the degradation is only 17%, due to the increase of parallel communications.

Finally, the scaling with plasma size has been checked by running 10 iterations on 16,000 cores for plasma sizes ranging from  $\rho^*=1/75$  to  $\rho^*=1/600$ . The time per iteration for a fixed time step scales like  $\rho^{*-3}$  as can be anticipated theoretically. However, in practice a simulation should be run up to  $t_{\text{fin}} \propto ca/c_s \propto \rho^{*-1}$ , such that practical simulations will scale like  $\rho^{*-4}$ .

#### IV. The Straight-Field-Line Solver

The memory of the field solver described in Section III scales as  $(\rho^*)^3$ . For standard simulations,  $\rho^*=1/150$  and the needed memory is 780 Mb. For ITER plasmas,  $\rho^*=1/1000$  and 165 Gb would be required. Clearly, a new solver is needed. The straight-field-line solver presented in Reference 2 has been implemented in GT5D. First, the quasi-neutrality linear system  $Ax=B$  is transformed into an equivalent system  $Cy=D$  with  $C=FAF^{-1}$ ,  $y=Fx$ ,  $D=FB$  and  $F$  is the Discrete Fourier Transform (DFT) operator in the



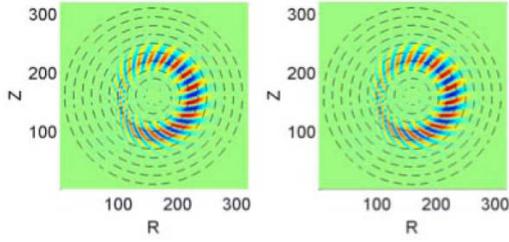
**Fig. 2** Weak scaling speedup achieved with GT5D on a Fujitsu BX900 machine from 2,000 to 16,000 cores.

poloidal space. Any poloidal coordinate may be used. A convenient choice is to use the straight-field-line angle  $\theta_*$  defined by:

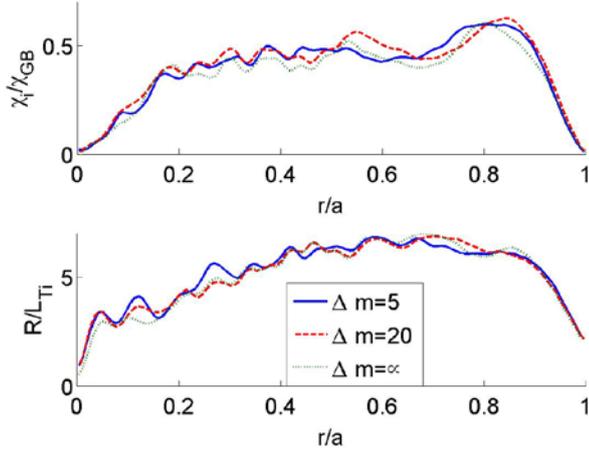
$$\theta_* = \frac{1}{q(r)} \int_0^\theta \frac{\mathbf{B} \cdot \nabla \varphi}{\mathbf{B} \cdot \nabla \theta'} d\theta' \quad (6)$$

Where  $q(r)$  is the safety factor. This coordinate is such that magnetic field lines are straight in a  $(\theta_*, \varphi)$  plane. This choice is adapted because the turbulence aligns with the magnetic field lines: one of the main assumptions of the gyrokinetic theory is the smallness of the ratio  $k_{\parallel}/k_{\perp}$ . Large  $k_{\parallel}$  components are physically damped by Landau damping. Consequently, the turbulent Fourier spectrum of the turbulence is extremely narrow when  $\theta_*$  is used. Since  $k_{\parallel} = (m-nq(\psi))/q(\psi)R$ , it means that relevant Fourier modes are given by  $|m-nq(\psi)| < \Delta m$ , with  $\Delta m$  a small integer number (typically 5). The idea of the straight-field-line solver is then to filter out Fourier modes that are not satisfying this inequality. Each processor must now store  $3N_r(2\Delta m+1)^2 N_{\varphi} / 4P_{\varphi} \propto (\rho^*)^{-1}$  complex elements, where  $N_{\varphi}$  is the number of grid points in the toroidal direction and  $P_{\varphi}$  is the number of processors used to parallelize the solver in the toroidal direction. Remarkably, the value of  $\Delta m$  neither depends on the plasma size nor on the magnetic geometry provided  $\theta_*$  is used as the poloidal coordinate. An ITER plasma would require only 80 Mb. The Fourier filter depends on the magnetic surface through the safety factor profile. It has been checked that this procedure does not affect the numerical particle and energy conservation properties of the simulations.

**Figure 3** shows the poloidal cross-sections of the potential for linear simulations with the original and straight-field-line solvers. They are undistinguishable, showing that  $\Delta m=3$  is enough to capture linear physics of ITG turbulence. Growth rates differ by less than 1%. In non-linear simulations, it must be checked that the filtering procedure does not affect the steady state of the simulation. In ITG turbulent simulations, Coulomb collision processes (resp. the perturbed electric field) induce neoclassical (resp.



**Fig. 3** Poloidal cross-section of the potential for the original solver (left) and the straight-field-line solver with  $\Delta m=3$ . Axes unit is  $\rho_s$



**Fig. 4** Radial profile of  $\chi_i/\chi_{GB}$  (top) and  $R_0/L_{Ti}$  (bottom) for GT5D simulations with different  $\Delta m$  values

turbulent) radial fluxes. For example, the radial turbulent heat flux is defined by:

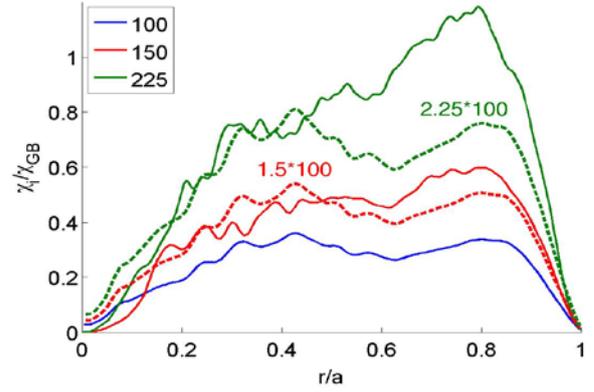
$$Q_i(r, t) = \left\langle \int d^3v f(\mathbf{R}, \mathbf{v}, t) \frac{1}{2} m_i v^2 \mathbf{v}_{E \times B} \cdot \frac{\nabla r}{|\nabla r|} \right\rangle_f. \quad (7)$$

The associate transport coefficient is defined by:

$$\chi_i = -\frac{n_i Q_i}{\nabla T_i} \propto \frac{\Delta x^2}{\Delta t} \quad (8)$$

GT5D is a fixed-flux code: the source operator  $S_{src}$  inputs some heat in the  $0 < r/a < 0.5$  region, which is transported radially and absorbed by the sink operator in the  $0.8 < r/a < 1$  region. Like in the experiments, the main output of the code is the profiles (density, parallel momentum and temperature), which are dictated by the input power and by plasma turbulence.

**Figure 4** shows the radial profiles of  $\chi_i/\chi_{GB}$  and  $R/L_{Ti}$ , where  $\chi_{GB}=(\rho^*)^2 c_s a$  is a normalization coefficient and  $L_{Ti}=T_i/|\nabla T_i|$  is the characteristic length of temperature for simulations with different values of  $\Delta m$ , averaged over the steady-state phase of the simulations.  $\Delta m = \infty$  is run with the Fourier solver and no filtering. This case is equivalent to the original solver (up to machine precision). The parallel filtering does not significantly affect the steady state of the simulations. The zonal flow amplitude differ maximum by 5% and differences up to 15% are observed for  $\chi_i/\chi_{GB}$ , which are comparable to the intrinsic variability observed in GT5D simulations. In other words, due to its chaotic behavior, plasma turbulence is sensitive to initial conditions: this sen-

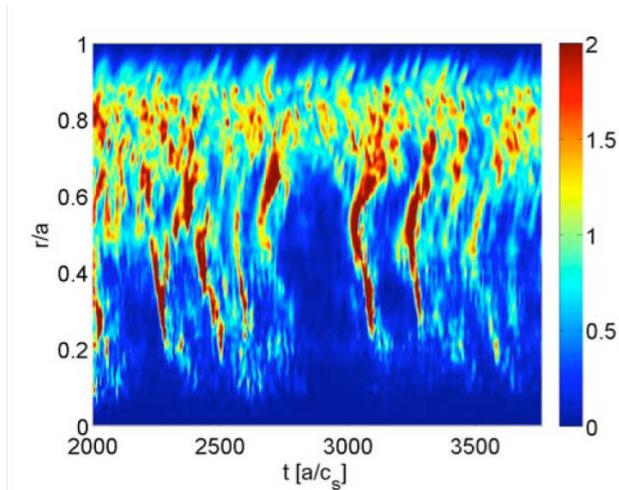


**Fig. 5** Radial profile of time-averaged  $\chi_i/\chi_{GB}$  for three GT5D simulations at  $\rho^*=1/100, 1/150$  and  $1/225$ . Dashed-lines show a Bohm-like scaling.

sitivity decrease (in the average sense) when the simulation time is increased, but it can be as high as 15% for the simulation times considered here. Since the effects of parallel filtering are smaller, or at most comparable to the effects of initial conditions, they can be considered to be not significant.

## V. $\rho^*$ Scaling

Plasma size is one of the main plasma parameters affecting fusion performance. For example, the Kadomstev constraint<sup>8)</sup> leads to  $B \tau_E \sim (\rho^*)^{-2.7}$ . This strong dependence is one of the reasons why future devices will have a larger minor radius. The confinement time will also depend on the turbulent activity in the plasma, but this dependence is yet still unknown. However, since ITER will have  $\rho^*$  twice as small as present day tokamaks, the understanding of plasma size on plasma turbulence is of primary importance. Heat transport is characterized through the relation  $\chi_i/\chi_B \propto (\rho^*)^{-\alpha}$  where  $\chi_B=(\rho^*)^{-1} \chi_{GB}$ . If  $\alpha = 1$ , transport is said to be Gyro-bohm; if  $\alpha = 0$ , transport is said to be Bohm. Finally, if  $\alpha < 0$  transport is said to be worse-than-Bohm. Many scaling laws assume Gyrobohm scaling, but both experiments<sup>9)</sup> and fluid simulations<sup>10)</sup> have reported Bohm scaling. Fixed-gradient gyrokinetic simulations find a transition from Bohm to Gyro-Bohm scaling when  $\rho^*$  decreases which depends on an effective  $\rho^*$  defined as  $\rho^*_{eff} = \sqrt{(\tau) \rho_{Li}/w}$  where  $w$  is the temperature profile gradient width, indicating that finite size effects on transport might still be important for ITER<sup>11)</sup>. However, fixed flux simulations do not fix the profiles and the effects of plasma size for such physical model must be studied. This becomes now possible with the straight-field-line solver implemented in GT5D. Simulations at  $\rho^*=1/100, 1/150$  and  $1/225$  have been performed by varying plasma size and keeping the total input power fixed (2 MW). The latter simulation has required  $1.4 \cdot 10^6$  CPU hours on the JAEA BX900 cluster. Assuming a direct relation between the input and turbulent heat flux, transport should be Gyro-Bohm like. **Figure 5** shows the time-averaged profile of  $\chi_i/\chi_{GB}$  for the 3 different simulations, which exhibit worse-than-Bohm scaling in the source-free region  $0.5 < r/a < 0.8$ . Worse-than-Bohm scaling has already been observed for extremely steep profiles<sup>12)</sup> or



**Fig. 6** Radial and temporal evolution of  $\chi_i/\chi_{GB}$  for the  $\rho^*=1/225$  simulation

very small plasma sizes.<sup>11)</sup> It is observed for the first time in experimentally realistic conditions in this work. Indeed, as displayed on **Fig. 6**, GT5D simulations exhibit heat avalanches which propagate inwards or outwards depending on the sign of the radial electric shear.<sup>1)</sup> This non-local processes may be responsible for the observed worse-than-Bohm scaling. The radial electric field profile is linked with the parallel momentum profile through the force balance relation.<sup>1)</sup> Such complicated inter-dependence is yet still not understood.

## VI. Conclusions

In this work, the implementation of a new straight-field-line solver in the GT5D code is presented. Global gyrokinetic codes are the most advanced tools to study turbulence in magnetically confined plasmas and require intensive optimizations and top-end HPC resources. GT5D has good scalability up to 16,000 processors, using an MPI-openMP hybrid parallelization scheme.

The limiting solver in terms of memory has been replaced with a new scheme that takes advantage of the alignment of turbulence with the magnetic field lines by filtering small parallel wavelengths of the turbulent spectrum. For large plasma size, at least one order of magnitude is gained in the total memory, and the simulation results are unaffected by the Fourier filtering. Then, the code is applied to study the influence of plasma size on ITG turbulence. Results show a

worse-than-Bohm scaling in the source-free region. Larger plasma size simulations will be required to assess the  $\rho^*$  effects on ITER.

Future developments will include the kinetic electron response and the extension to multiple ion species.

## Acknowledgment

The simulations presented in this work were performed on the JAEA Altix3700Bx2 system and on the JAEA BX900 system. One of the authors (Y. I.) is supported by the MEXT, Grant No. 22866086.

## References

- 1) Y. Idomura *et al.*, “Study of ion turbulent transport and profile formations using global gyrokinetic full- $f$  Vlasov simulations,” *Nucl. Fusion*, **49**, 065029 (2009).
- 2) B. F. McMillan *et al.*, “Rapid Fourier space solution of linear partial integro-differential equations in magnetic confinement geometries,” *Comput. Phys. Commun.*, **181**, 715 (2010).
- 3) Y. Idomura *et al.*, “Conservative global gyrokinetic toroidal full- $f$  five dimensional Vlasov simulation,” *Comput. Phys. Commun.*, **179**, 391 (2008).
- 4) X. Q. Xu, M. N. Rosenbluth, “Numerical simulation of ion-temperature-gradient-driven modes,” *Phys. Fluids*, **B3**, 627 (1991).
- 5) S. Satake *et al.*, “Benchmark test of drift-kinetic and gyrokinetic codes through neoclassical transport simulations,” *Comput. Phys. Commun.*, **181**, 1069 (2010).
- 6) X. Zhong, “Additive Semi-Implicit Runge-Kutta Methods for Computing High-Speed Nonequilibrium Reactive Flows,” *J. Comput. Phys.*, **128**, 19 (1996).
- 7) B. F. Mcmillan *et al.*, “System size effects on gyrokinetic turbulence,” *Phys. Rev. Lett.*, **105**, 155001 (2010).
- 8) B. B. Kadomstev, “Tokamaks and dimensional analysis,” *Sov. J. Plasma Phys.*, **1**, 295 (1975).
- 9) C. C. Petty *et al.*, “Gyroradius scaling of Electron and Ion Transport,” *Phys. Rev. Lett.*, **74**[10], 1763 (1995).
- 10) X. Garbet, R. E. Waltz, “Action at distance and Bohm scaling of turbulence in tokamaks,” *Phys. Plasmas*, **3**[5], 1898 (1996).
- 11) L. Villard *et al.*, “Gyrokinetic simulations of turbulent transport: size scaling and chaotic behaviour,” *Plasma Phys. Control. Fusion*, **52**, 124038 (2010).
- 12) R. E. Waltz *et al.*, “Gyrokinetic turbulence simulations of profile shear stabilization and broken gyroBohm scaling,” *Phys. Plasmas*, **9**[5], 1938 (2002).