Monte Carlo Simulation Code for Solving Radial Fluid Equations in Toroidal Plasmas[*)]

Ryutaro KANNO[1,2)], Shinsuke SATAKE[1,2)] and Masanori NUNAMI[1)]

[1)]*National Institute for Fusion Science, Toki 509-5292, Japan*

[2)]*Department of Fusion Science, the Graduate University for Advanced Studies, Toki 509-5292, Japan*

(Received 6 December 2010 / Accepted 19 April 2011)

We develop a new Monte Carlo simulation method to calculate steady-state solutions of fluid equations for edge plasmas. To confirm the computational principle of the new method, benchmark tests including nonlinear problems in one dimensional (i.e., radial) coordinate space are attempted in the first trial; the code based on the method is called DIPS-1D. We confirm that DIPS-1D is useful for solving a Dirichlet problem and that the solution given by the present method provides sufficient numerical accuracy.

Keywords: Monte Carlo method, fluid equation, Dirichlet problem, stochastic analysis, diffusion process

DOI: 10.1585/pfr.6.2403066

1. Introduction

Studies of edge transport in toroidal plasmas are based mostly on Braginskii's fluid equations [1, 2], which are given as

$$\frac{\partial u(t,\boldsymbol{x})}{\partial t} + \nabla\cdot(\boldsymbol{U}u) - \frac{1}{2}\nabla\cdot(\mathsf{D}\cdot\nabla u) + \eta u = h(t,\boldsymbol{x}), \tag{1}$$

for a fluid quantity $u(t,\boldsymbol{x})$, i.e. the density $u = n$, the parallel momentum $u = p_{\|}$ and the energy $u = (3/2)nT$ where T is the temperature, in three dimensional (3D) Euclidean space (coordinate space) $\boldsymbol{R}^3$, with the initial condition $u(0,\boldsymbol{x}) = \Phi(\boldsymbol{x})$ at $t = 0$ and the boundary condition $u(t,\boldsymbol{x}) = G(t,\boldsymbol{x})$ on the boundary. The coordinates are defined as $\boldsymbol{x} = {}^{\mathrm{t}}(x^1,x^2,x^3) \in \boldsymbol{R}^3$, $\nabla = \partial/\partial\boldsymbol{x}$, and $t \in [0,t_1)$ is time. $\boldsymbol{U}(t,\boldsymbol{x}) = {}^{\mathrm{t}}(U^1,U^2,U^3)$ represents the convection velocity term, and η is assumed to be lower bounded and is interpreted as the so-called *killing rate* if $\eta > 0$ and the *birth rate* if $\eta < 0$. $\mathsf{D}(t,\boldsymbol{x}) = (D^{ij})$ is the diffusion matrix $(i,j = 1,2,3)$. If the edge plasma exists in an ergodic region, the fluid equations are usually solved by Monte Carlo methods [3, 4] because the profiles of the fluid quantities are significantly affected by the complex structure of the magnetic field lines including magnetic islands and chaotic field lines [5]. However, the Monte Carlo methods used in previous studies could not easily solve the time evolution of the fluid equations because of nonlinear terms, as shown in Ref. [6]. In addition, the fluid representation has an intrinsic disadvantage in that diffusion coefficients are generally not determined only in the fluid equations. The simplest and best way to overcome these difficulties might be full-kinetic-particle simulations (so-called full-f simulations) [7], where f is the total distribution function of the particles. However, full-f simulations are generally time-consuming and recent study showed that the scales of the time evolution of f_{M} and δf are well separated even in the ergodic region [8]. The total distribution function f is given as $f = f_{\mathrm{M}} + \delta f$, where f_{M} is a Maxwellian distribution function of the background plasma described by the fluid equations, and δf is a small deviation (i.e., $\delta f = f - f_{\mathrm{M}}$) determining the diffusion coefficients. As a result, rather than conducting full-f simulations, we should separate the treatments of the fluid quantities and diffusion coefficients [8–10] and consider improving the Monte Carlo methods used to solve the fluid equations.

In this paper, we report a new Monte Carlo simulation code produced in the beginning of our code development for overcoming the difficulty caused by nonlinear terms in the fluid equations and for calculating the steady-state solutions. To confirm the computational principle of the new method, we solve Dirichlet problems in one dimensional (1D, i.e., radial) coordinate space in the first trial. The code is called DIPS-1D (DIrichlet Problem Solver in 1D coordinate space). The details of the Monte Carlo method are presented in Sec. 2. Sec. 3 describes benchmark tests including nonlinear problems that the method successfully passed. Finally, a summary is given in Sec. 4.

2. Monte Carlo Method for Solving Fokker-Planck Type Equations

The fluid equation expressed in the form of the Fokker-Planck type equation (1) is rewritten as the following initial-boundary value problem (t is replaced by $t_1 - t$):

$$(L + \eta_*)u + \frac{\partial u}{\partial t} = h_*(t,\boldsymbol{x}) \text{ in } Q, \tag{2}$$

$$u(t_1,\boldsymbol{x}) = \Phi(\boldsymbol{x}) \text{ on } \mathcal{M}, \tag{3}$$

$$u(t,\boldsymbol{x}) = G(t,\boldsymbol{x}) \text{ on } \mathcal{S}, \tag{4}$$

author's e-mail: kanno@nifs.ac.jp

[*)] This article is based on the presentation at the 20th International Toki Conference (ITC20).

where $\mathcal{M}$ is a bounded domain with the boundary $\partial\mathcal{M}$, $Q = \mathcal{M} \times [0, t_1)$, $\mathcal{S} = \partial\mathcal{M} \times [0, t_1)$, and

$$Lu = \left\{ \frac{1}{2} D^{ij} \frac{\partial^2}{\partial x^i \partial x^j} + U_*^i \frac{\partial}{\partial x^i} \right\} u, \tag{5}$$

$$U_*^i = -U^i + \frac{1}{2} \frac{\partial D^{ij}}{\partial x^j}, \tag{6}$$

$$\eta_* = -\eta - \frac{\partial U^i}{\partial x^i}, \tag{7}$$

$$h_* = -h. \tag{8}$$

If Φ, G, h, η, $\boldsymbol{D}$ and $\boldsymbol{U}$ are assumed to be given-smooth-functions, the solutions of Eqs. (2)-(4) are known to be described as [11–13]

$$\begin{aligned} & u(t, \boldsymbol{x}\,; \Phi, G, h_*, \eta_*, \boldsymbol{D}, \boldsymbol{U}_*) \\ & \quad = E_{t,\boldsymbol{x}}\Big[\Phi(\boldsymbol{X}(t_1)) \exp\Big\{ \int_t^\tau \eta_*(s, \boldsymbol{X}(s)) \mathrm{d}s \Big\} \chi_{\tau = t_1}\Big] \\ & \qquad + E_{t,\boldsymbol{x}}\Big[G(\tau, \boldsymbol{X}(\tau)) \exp\Big\{ \int_t^\tau \eta_*(s, \boldsymbol{X}(s)) \mathrm{d}s \Big\} \chi_{\tau < t_1}\Big] \\ & \qquad - E_{t,\boldsymbol{x}}\Big[\int_t^\tau h_*(s, \boldsymbol{X}(s)) \exp\Big\{ \int_t^s \eta_*(\vartheta, \boldsymbol{X}(\vartheta)) \mathrm{d}\vartheta \Big\} \mathrm{d}s \Big], \end{aligned} \tag{9}$$

where $E_{t,\boldsymbol{x}}$ is the expectation operator given by the diffusion process $\boldsymbol{X}(s)$ in coordinate space:

$$\mathrm{d}X^i(s) = \sigma^i_j(t, \boldsymbol{X}(s)) \mathrm{d}W^j(s) + U_*^i(t, \boldsymbol{X}(s)) \mathrm{d}s, \tag{10}$$

satisfying $\boldsymbol{X}(t) = \boldsymbol{x}$. Here $D^{ij} = \sigma^i_k \mathrm{g}^{k\ell} \sigma^j_\ell$, $\mathrm{g}^{k\ell}$ is the metric, $\boldsymbol{W}(s)$ is a Brownian process, χ_A is the indicator function of a set A (e.g., $\chi_{\tau<t_1} = 1$ if $\tau < t_1$, and $\chi_{\tau<t_1} = 0$ otherwise), and τ is defined as

$$\tau = \begin{cases} \text{the first time } \vartheta \in [t, t_1) \text{ that } \boldsymbol{X}(\vartheta) \text{ leaves } \mathcal{M} \\ \qquad\qquad\qquad\qquad \text{if such a time exists,} \\ t_1 \ \text{ otherwise;} \end{cases}$$

see Refs. [11, 12]. Hereafter, we consider the fluid equations in 1D coordinate space.

The simulation code DIPS-1D based on Eq. (9) with sufficiently large t_1 satisfying $\chi_{\tau=t_1} = 0$ and $\chi_{\tau<t_1} = 1$ for all the random walkers used in the code is programmed by the following scheme.

(i) The initial guess $u^{(0)}(0, x)$ is assumed to be the initial condition itself $\Phi(x)$, and a temporary solution at the first step $u^{(1)}(0, x\,; \Phi^{(0)}, G^{(0)}, h_*^{(0)}, \eta_*^{(0)}, \boldsymbol{D}^{(0)}, \boldsymbol{U}_*^{(0)})$ is given by Eq. (9) with $\Phi^{(0)}$, $G^{(0)}$, $h_*^{(0)}$, $\eta_*^{(0)}$, $\boldsymbol{D}^{(0)}$, and $\boldsymbol{U}_*^{(0)}$, which are evaluated using $u^{(0)}(0, x) = \Phi^{(0)}(x) = \Phi(x)$.

(ii) After calculating $u^{(k)}(0, x)$ at the kth step, a temporary solution at the $(k+1)$th step $u^{(k+1)}(0, x\,; \Phi^{(k)}, G^{(k)}, h_*^{(k)}, \eta_*^{(k)}, \boldsymbol{D}^{(k)}, \boldsymbol{U}_*^{(k)})$ is given by Eq. (9) with $\Phi^{(k)}$, $G^{(k)}$, $h_*^{(k)}$, $\eta_*^{(k)}$, $\boldsymbol{D}^{(k)}$, and $\boldsymbol{U}_*^{(k)}$, which are evaluated using $u^{(k)}(0, x) = \Phi^{(k)}(x)$, where $k = 1, 2, 3, \ldots$.

(iii) The error at the $(k+1)$th step, $\epsilon(k+1)$, is estimated as

$$\epsilon(k+1) = \frac{\int_{x_1}^{x_2} \mathrm{d}x \left| u^{(k+1)}(0, x) - u^{(k)}(0, x) \right|}{\int_{x_1}^{x_2} \mathrm{d}x \left| u^{(k)}(0, x) \right|}. \tag{11}$$

If $\epsilon(k+1)$ is less than ϵ_0, then $u^{(k+1)}(0, x)$ is considered to be a steady-state solution satisfying Eq. (1), where ϵ_0 (> 0) is a sufficiently small real number, x_1 and x_2 are the positions of the boundaries in 1D space, and $x_1 < x_2$ is assumed.

A result given by the Monte Carlo method generally has numerical noise caused by random walkers themselves generating sample paths $\{X(s)\}$, where sample paths $\{X(s)\}$ described by Eq. (10) are calculated by the second-order Runge-Kutta scheme in the code. To smooth the Monte Carlo result, we used the polynomial regression model based on the Akaike information criterion (AIC) [14] in this paper. The smoothed temporary solution, which is differentiable as distinct from the Monte Carlo result itself, is expressed as $u^{(k)}(0, x)$ in scheme (i)-(iii). According to this scheme, a steady-state solution satisfying Eq. (1) is found iteratively. Note that the above scheme is completely different from the computational schemes in Refs. [3, 4] which are based on numerical techniques using a transition probability of each fluid quantity and its convolution. The Monte Carlo method in this section is analogous to the Galerkin method. In our method, however, a temporary solution approximately satisfying Eq. (1) is found directly by Eq. (9) without any basis function.

3. Benchmarks of DIPS-1D

In this section, we attempt to confirm the validity of DIPS-1D by using several benchmark tests. First, the following Dirichlet problem (*test A*) is considered.

$$\begin{aligned} & \left(x^3 + 8x^2 + \frac{1}{2}x \right) \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} - (4 + x) \frac{\mathrm{d}u}{\mathrm{d}x} - 2(1 + x)u \\ & \quad = 5x(1 - x) - 2x. \end{aligned} \tag{12}$$

The initial and boundary conditions are $\Phi(x) = 1 - x$, $G(t, 0) = 1$, and $G(t, 1) = 0$, where $h_* = 5x(1-x) - 2x$, $\eta_* = -2(1+x)$, $D = x^3 + 8x^2 + (1/2)x$, and $U_* = -(4+x)$. The solution is given analytically as $u(0, x) = 1 - (1/2)x - (1/2)x^2$. The solution found by DIPS-1D agrees with the analytical solution, as shown in Fig. 1. When we change h_* and η_* in test A to $h_* = 5x(1 - x) - 2x + 2(1 + x)u(t, x)$ and $\eta_* = 0$ (*test B*), the solution found by DIPS-1D also agrees with the analytical solution, as shown in Fig. 1.

From the result of test B, we find that the term $\eta_* u$ can be interpreted as a source (or sink) term in Eq. (9) in the iterative calculation of a steady-state solution. This numerical technique is especially useful for cases of $\eta_* > 0$ because it suppresses numerical errors in the integrals in Eq. (9), as shown, for example, in Fig. 2. The control parameters of the code in Fig. 2 (e.g., the evolutionary time step, and total number of random walkers, except for the treatment of h_* and η_*) are fixed in both cases.

Next, we consider the Burgers equation [15, 16].

$$\frac{\partial u}{\partial t} - u \frac{\partial u}{\partial x} + \frac{D}{2} \frac{\partial^2 u}{\partial x^2} = 0, \tag{13}$$

where $D/2 = 0.05$ is a constant viscosity, and the initial and boundary conditions are given as $\Phi(x) = 1 - 2x$,

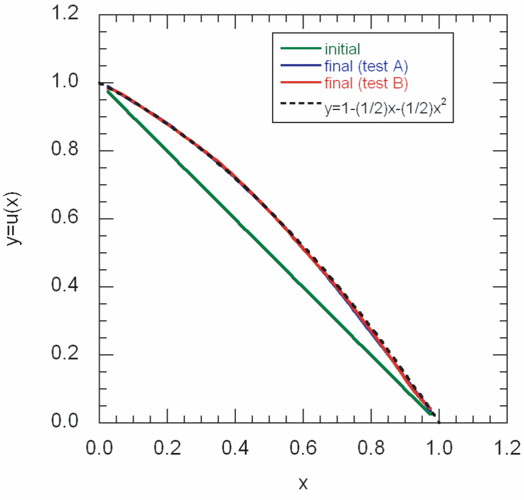

Fig. 1 Benchmark tests of solving Eq. (12). Dashed black line represents the analytical solution $u(0, x) = 1 - (1/2)x - (1/2)x^2$, where the boundary condition is $G(t, 0) = 1$ and $G(t, 1) = 0$. Solid green line represents the initial guess $u^{(0)}(0, x) = \Phi(x) = 1 - x$. Solid blue and red lines represent solutions found by DIPS-1D for test A and test B, respectively. Here, $\epsilon < \epsilon_0 = 1/200$ is satisfied in both tests.

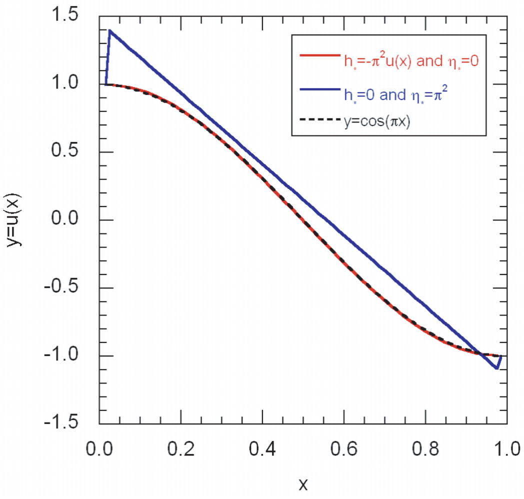

Fig. 2 Benchmark test of solving $\mathrm{d}^2u(x)/\mathrm{d}x^2 + \pi^2 u(x) = 0$ with the initial and boundary conditions $u^{(0)}(0, x) = \Phi(x) = -2x + 1$, and $G(t, 0) = 1$, $G(t, 1) = -1$, respectively. Dashed black line represents the analytical solution $u(0, x) = \cos(\pi x)$. Solid red line represents the result when $h_* = -\pi^2 u$ and $\eta_* = 0$, where $\epsilon < \epsilon_0 = 1/100$. Solid blue line represents the result when $h_* = 0$ and $\eta_* = \pi^2$ without smoothing and is a failure in searching for the solution ($\epsilon \gg \epsilon_0 = 1/100$).

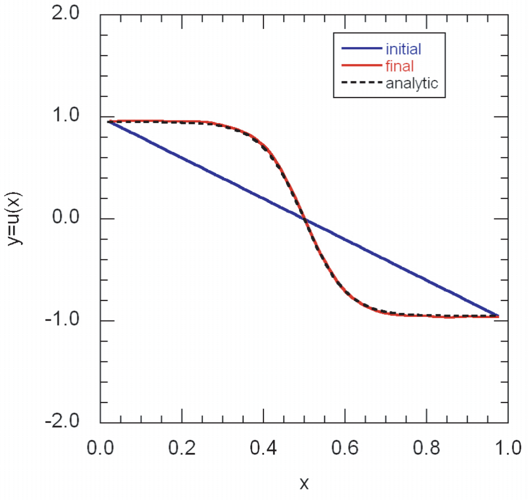

Fig. 3 Benchmark test of solving the Burgers equation, where the boundary condition is $G(t, 0.025) = 0.95$ and $G(t, 0.975) = -0.95$. Solid blue line represents the initial guess $u^{(0)}(0, x) = \Phi(x)$. Solid red line represents the solution found by DIPS-1D. Dashed black line represents the analytical solution. Here, $\epsilon < \epsilon_0 = 1/100$ is satisfied.

$G(t, 0.025) = 0.95$, and $G(t, 0.975) = -0.95$. Here, t is replaced by $t_1 - t$ in Eq. (13). We find that the steady-state solution found by DIPS-1D agrees with the analytical solution given approximately as $u(0, x) = -0.95 \tanh\{9.5(x - 0.5)\}$, as shown in Fig. 3. Thus, the Monte Carlo simulation code DIPS-1D is valid for finding a steady-state solution to such a nonlinear equation.

Finally, the radial energy transport for electrons [2] is considered (in the equation below, t is replaced by $t_1 - t$).

$$\frac{\partial}{\partial t}\left(\frac{3}{2} n T_\mathrm{e}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r \kappa_\mathrm{e}^{gB} \frac{\partial T_\mathrm{e}}{\partial r}\right) - \frac{3}{2}\langle \sigma v \rangle_\mathrm{rre} n^2 T_\mathrm{e} + S = 0, \tag{14}$$

where the initial condition is given as $\Phi(r) = u^{(0)}(0, r) = T_\mathrm{ax}^{(0)} - \left\{T_\mathrm{ax}^{(0)} - T_\mathrm{edge}^{(0)}\right\} r/a$ with $T_\mathrm{ax}^{(0)} = 2$ keV and $T_\mathrm{edge}^{(0)} = 200$ eV. The boundary conditions are $G(t, r_0) = T_\mathrm{e}(t, r_0) = T_\mathrm{e}(t, r_0 + \delta r) - \delta r\, \partial T_\mathrm{e}/\partial r(t, r_0)$ and $G(t, a) = T_\mathrm{e}(t, a) = 200$ eV with sufficiently small δr and r_0 (i.e., $0 < \delta r/a$, $r_0/a \ll 1$), $T_\mathrm{e}(t, r_0 + \delta r) = u^{(k-1)}(0, r_0 + \delta r)$ with $k = 1, 2, 3, \ldots$, and $\partial T_\mathrm{e}/\partial r(t, r_0) = -1.8$ keV/m. T_e is the electron temperature, and the constant density is assumed to be $n = 1 \times 10^{19}$ m^{-3}. To reduce the computation time, we employed the gyro-Bohm thermal conductivity $\kappa_\mathrm{e}^{gB} = n(T_\mathrm{e}/eB)(\rho_\mathrm{e}/\beta q a)(r/R)^2$ m^2/s [17] in the diffusion term of Eq. (14) instead of the conductivity given by the drift kinetic equation of δf. Here, $R = 3$ m and $a = 1$ m are the major and minor radii, respectively, B is the magnetic field strength with $B_\mathrm{ax} = 10$ T at the axis, ρ_e is the electron gyroradius, β is the beta value, and $q = 1/\{0.9 - 0.5875(r/a)^2\}$ is the safety factor. We consider the effects of radiative recombination and the heat source in this benchmark test, where $\langle \sigma v \rangle_\mathrm{rre} = 1.27 \times 10^{-19} (I/T_\mathrm{e})^{3/2}/\{(I/T_\mathrm{e}) + 0.59\}$ m^3/s is the radiative recombination rate coefficient with $I = 13.6$ eV [18], and $S = (5/2)\{1 - 4(r/a - 1/2)^2\}$ kW/m^3 is the heat source. Equation (14) is a nonlinear equation; i.e., κ_e^{gB} and $\langle \sigma v \rangle_\mathrm{rre}$ are functions of T_e itself.

Note that the boundary condition at $r = r_0$, $G(t, r_0)$, is

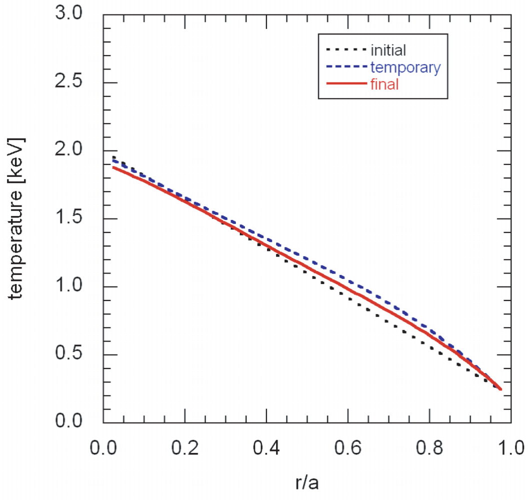

Fig. 4 Benchmark test of radial energy transport for electrons with the boundary condition is $G(t, r_0) = T_e(t, r_0 + \delta r) - \delta r\, \partial T_e/\partial r(t, r_0)$ and $G(t, a) = \text{constant} = 200\,\text{eV}$, where $\partial T_e/\partial r(t, r_0) = \text{constant} = -1.8\,\text{keV/m}$. Dotted black line represents the initial guess $u^{(0)}(0, r) = \Phi(r)$. Dashed blue line represents the solution found by DIPS-1D at the temporary step with $k = 1$ (i.e., the first step), and solid red line represents the solution at the final step with $\epsilon < \epsilon_0 = 1/100$.

not constant and is set to satisfy $\partial T_e/\partial r(t, r_0) = \text{constant}$ in this benchmark test; i.e., the boundary condition given by the derivative $\partial T_e/\partial r$ is executable in the DIPS-1D scheme. DIPS-1D is designed to solve Dirichlet problems, but its iterative scheme renders treatment of Neumann and/or Robin boundary conditions technically feasible.

The steady-state solution of Eq. (14) is given in Fig. 4. We confirm that the steady-state solution given by DIPS-1D in Fig. 4 does not depend on the total number of random walkers, the initial guesses, or the random number sequences, as shown in Figs. 5 and 6. Note that the numerical accuracy of simulation results calculated by Monte Carlo methods generally depends on the total number of random walkers. As shown in Fig. 5, the solution at the final step in Fig. 4 converges sufficiently. In Fig. 6, all the initial guesses are chosen to be analytic functions that are not far from the solution at the final step satisfying $\epsilon < \epsilon_0$. In this paper, pseudo random numbers are generated by Tausworthe sequences, and physical random numbers are generated by the physical random number generator on the supercomputer *HITACHI SR16000*. The use of physical random numbers is efficient in the numerical search for an appropriate steady-state solution and in confirming the independence of the simulation results from the generated random numbers because it is extremely rare for the physical random number generator to generate the same random number sequence twice in a row. The benchmark tests in Fig. 6 clearly validate the Monte Carlo method presented here.

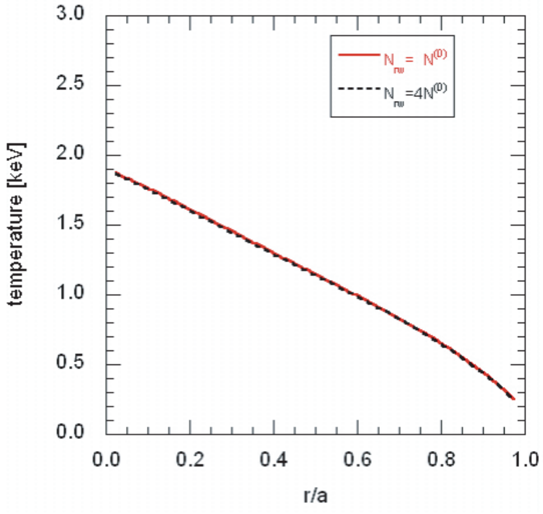

Fig. 5 Dependence of the result at the final step in Fig. 4 on the total number of random walkers N_{rw}. Solid red line represents the solution at the final step if the total number in Fig. 4 is $N_{\text{rw}} = N^{(0)} = 101\,000$. Dashed black line represents the solution if $N_{\text{rw}} = 4N^{(0)}$. Here, $\epsilon < \epsilon_0 = 1/100$ is satisfied in both cases.

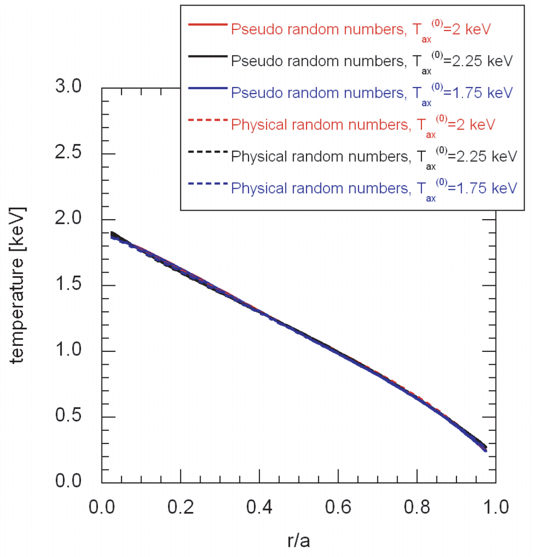

Fig. 6 Dependence of the result at the final step in Fig. 4 on the initial guesses $u^{(0)}(0, r) = T^{(0)}_{\text{ax}} - \{T^{(0)}_{\text{ax}} - T^{(0)}_{\text{edge}}\}r/a$ with $T^{(0)}_{\text{edge}} = 200\,\text{eV}$ and random number generators. Solid red line represents the solution at the final step if $T^{(0)}_{\text{ax}} = 2\,\text{keV}$ (similar to Fig. 4) is applied. Solid black line represents the solution if $T^{(0)}_{\text{ax}} = 2.25\,\text{keV}$. Solid blue line represents the solution if $T^{(0)}_{\text{ax}} = 1.75\,\text{keV}$. In these cases, pseudo random numbers are used. In contrast, physical random numbers are used in the following cases. Dashed red line represents the solution if $T^{(0)}_{\text{ax}} = 2\,\text{keV}$. Dashed black line represents the solution if $T^{(0)}_{\text{ax}} = 2.25\,\text{keV}$. Dashed blue line represents the solution if $T^{(0)}_{\text{ax}} = 1.75\,\text{keV}$. In all cases, $\epsilon < \epsilon_0 = 1/100$ is satisfied.

4. Summary

In this paper, we develop a new Monte Carlo simulation method for solving fluid equations expressed in the Fokker-Planck-type form and for calculating the steady-state solutions. Several benchmark tests, including nonlinear problems in 1D coordinate space, are attempted in the first trial, and the validity of the computational principle of the new method for calculating steady-state solutions without any additional special-techniques against nonlinearity is confirmed. The search for solutions of three fluid quantities (i.e., n, $p_{\|}/mn$ and T, where m is the particle mass) is not simultaneously executed in this first report on our code development, and will be described elsewhere in the near future.

The use of physical random numbers is efficient in the numerical search for an appropriate solution. The validation of the Monte Carlo method proposed here is ensured by using a physical random number generator because of the outstanding statistical quality of the generated random numbers. Execution of Eq. (9) by using random walkers can easily be parallelized and is intrinsically suitable for parallel computation.

The random walkers can easily follow the details of a magnetic field line structure, and thus are useful for searching for an appropriate solution in complex magnetic structures, including magnetic islands and ergodic regions in a 3D magnetic field configuration. We plan to extend DIPS-1D to a simulation code in 3D space for solving steady-state transport in edge plasmas. The code in 3D space can be programmed in the typical Eulerian coordinates and does not need a particular coordinate system. The computational principle of the code in 3D space is basically the same as that of DIPS-1D.

Acknowledgements

The computational simulations in this paper were conducted using the supercomputers (HITACHI SR16000) at National Institute for Fusion Science (NIFS) and National Institute of Polar Research (NIPR). We are grateful to Dr. Masaki Okada (NIPR) for giving the permission to use the physical random number generators in the NIPR supercomputer. We also thank Dr. Hisanori Takamaru (Chubu Univ.) for the permission to use his code for generating Tausworthe sequences. This work was performed with the support and under the auspices of the NIFS Collaborative Research Program NIFS10KTAT042 and NIFS09KNXN159.

[1] S.I. Braginskii, *Reviews of Plasma Physics, vol.1* (Consultants Bureau, New York, 1965).

[2] P.C. Stangeby, *The Plasma Boundary of Magnetic Fusion Devices* (IOP Pub., Bristol and Philadelphia, 2000).

[3] Y. Feng *et al.*, J. Nucl. Mater. **266-269**, 812 (1999).

[4] A. Runov *et al.*, Nucl. Fusion **44**, S74 (2004).

[5] M. Kobayashi *et al.*, J. Nucl. Mater. **363-365**, 294 (2007).

[6] R. Kanno *et al.*, J. Plasma Fusion Res. SERIES **6**, 527 (2004).

[7] W. Park *et al.*, Phys. Plasmas **6**, 1796 (1999).

[8] R. Kanno *et al.*, Plasma Phys. Control. Fusion **52**, 115004 (2010).

[9] S. Brunner *et al.*, Phys. Plasmas **6**, 4504 (1999).

[10] R. Kanno *et al.*, Contrib. Plasma Phys. **48**, 106 (2008).

[11] A. Friedman, *Stochastic Differential Equations and Applications* (Dover, New York, 2004).

[12] B. Øksendal, *Stochastic Differential Equations* (Springer-Verlag, Berlin Heidelberg, 2003).

[13] A.D. Wentzell, *A Course in the Theory of Stochastic Processes* (McGraw-Hill, New York, 1981).

[14] H. Akaike, in *2nd Inter. Symp. on Information Theory* (Akademiai Kiado, Budapest, 1973) pp.267-281.

[15] J.M. Burgers, Adv. Appl. Mech. **1**, 171 (1948).

[16] G.B. Whitham, *Linear and Nonlinear Waves* (John Wiely & Sons, New York, 1999).

[17] J. Wesson, *Tokamaks 2nd ed.* (Oxford Univ. Pr., New York, 1997).

[18] Yu. Gordeev *et al.*, Pis'ma Zh. Eksp. Teor. Fiz. **25**, 223 (1977); Gordeev *et al.*, JETP Lett. **25**, 204 (1977).