Monte Carlo Solver for Partly Calculating a Solution to the Poisson Equation in Three-Dimensional Curvilinear Coordinates

Ryutaro KANNO[1,2], Gakushi KAWAMURA[1,2], Masanori NUNAMI[1,3], Seikichi MATSUOKA[1,2] and Shinsuke SATAKE[1,2]

[1)]*National Institute for Fusion Science, National Institutes of Natural Sciences, Toki 509-5292, Japan*
[2)]*Department of Fusion Science, The Graduate University for Advanced Studies, SOKENDAI, Toki 509-5292, Japan*
[3)]*Nagoya University, Nagoya 464-8601, Japan*

(Received 21 November 2021 / Accepted 19 February 2022)

We develop a new simulation code for solving the Poisson equation, based on Monte Carlo methods. When static resonant magnetic perturbations (RMPs) are used in tokamak plasma to mitigate or suppress edge-localized modes, the RMPs generate an electric field in the ergodized edge region. The electrostatic potential should be calculated only in the edge region to reduce the computational cost of solving the Poisson equation in the complicated three-dimensional magnetic structure, which is assumed to be fixed in time. In this study, we propose a basic idea for evaluating an electrostatic potential given by the Poisson equation in only a part of the domain in curvilinear coordinates. This Poisson solver allows for the boundary condition to be set not only inside the selected region in which the potential is evaluated, but also outside the selected region. Several benchmarks for the developed code are also presented.

Keywords: Poisson solver, Monte Carlo method, curvilinear coordinate

DOI: 10.1585/pfr.17.1403029

1. Introduction

When numerical simulation methods are used to study impurity transport in the edge region of a quasi-steady tokamak plasma, the three-dimensionality of the magnetic field in the edge region should be considered. Currently, static resonant magnetic perturbations (RMPs) are commonly used to mitigate or suppress edge-localized modes (ELMs) in tokamak plasma [1–3]. Victor *et al.* [2] reported that "*RMPs are effective at controlling impurity build up in the pedestal region, regardless of ELM suppression*." Since the magnetic structure becomes three-dimensional (3D) in the edge region, the electric field generated in the 3D magnetic field is presumed to also affect the impurity transport. The radial electric field has been observed in the edge region ergodized by the RMPs in tokamak experiments [3]. In this study, we discuss how to evaluate the electrostatic potential in the ergodized edge region, where the vector potential is assumed to be fixed in time.

Typically, electrostatic potential is calculated by solving the Poisson equation. However, the Poisson solver's required performance is complex as the solver is applied to a highly nonuniform charge-density distribution, which is thought to be due to the complicated structure of magnetic field lines in the ergodized edge region [4]. A Poisson solver is often based on finite-difference or finite-element methods [5–7]. These techniques are commonly used for calculating the potential of the entire domain, such as the core, edge, peripheral, and vacuum regions, if the boundary condition is given at the vacuum vessel where the potential can be naturally set to zero. In addition, advanced techniques for constructing the mesh system are probably required when considering the complicated structure of magnetic field lines [7]. It is possible to calculate the potential even in the complicated structure by making the mesh size finer without using advanced techniques, but the computational cost increases in general and a more detailed calculation of the charge density is required. Considering the above discussion, when impurity transport in the edge region is simulated, the computational cost of evaluating the potential over the entire domain is high since the volume of the edge region is smaller than the entire domain within the vacuum vessel. Here, the cost includes the preparations for the potential calculation, as discussed above.

When focusing on the edge region where impurity transport is controlled, it is preferable to calculate the electrostatic potential only in the edge region, rather than for the entire domain. If a solver based on finite-difference or finite-element methods is applied only to the edge region, the values of the potential at the boundary, which should be set in the edge region, are required in advance for solving the Poisson equation. Alternatively, using the stochastic solution for the Poisson equation makes it possible to evaluate the potential values only in the edge region and reduce the computational cost, while maintaining the

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

boundary condition at the vacuum vessel. The stochastic solution gives a potential value at an arbitrary point in 3D coordinates. We can determine the potential values in a selected region within the entire domain using the numerical method based on the stochastic solution without requiring advanced techniques for the mesh system and defining the boundary condition in the selected region. In this numerical method, we treat separately the selected region and the region where the boundary condition is set. The idea of using the stochastic solution to numerically solve the Poisson equation has been proposed in previous studies, such as that by DeLaurentis and Romero [8]. In their study, the probabilistic algorithm consists of random "walk on spheres," where the spheres are finite (nonzero) in size and can be placed in any shaped domain. The stochastic solution is described using Green's function in polar coordinates. To limit the number of steps in the random walk, the walk is stopped whenever it reaches the boundary region Γ_δ, which is the boundary with a finite width, $\delta > 0$ [8]. When the solver is expanded to solve the Poisson equation in curvilinear coordinates and evaluate the potential in the edge region, it is effective to use the stochastic solution determined by random walks of point particles, i.e., sample paths, whose diffusion coefficient is twice the metric of the coordinates, as described in the study by Itô [9]. Here, the electrostatic potential can be evaluated by averaging the contribution of the point particles. Both the spheres and the boundary region, Γ_δ, are no longer required in the evaluation when using the point particles. Thus, we developed a new simulation code that solves the Poisson equation in curvilinear coordinates. The new code corresponds to an expansion of the Monte Carlo (MC) code DIPS-1D [10] into 3D curvilinear coordinates.

The following section of this paper discusses the new code using the theory of stochastic differential equations and presents several benchmarks for the developed code.

2. Monte Carlo Poisson Solver

The electrostatic potential, Φ, is given by the Poisson equation. The Poisson equation is expressed as follows:

$$\nabla^2\Phi(\mathbf{x}) = -\frac{1}{\epsilon_0}\rho(\mathbf{x}), \tag{1}$$

where ϵ_0 denotes the electric constant and ρ denotes the charge density. Since we are focused on impurity transport in the plasma edge, electrostatic potential, Φ, should be calculated only in the edge region, thereby reducing the computational cost. The MC simulation code for solving the Poisson equation, which is called DIPS-3D, is based on a mathematical result presented in Theorem 9.3.3 in Chapter 9 of Øksendal [11] and in Theorem 5.2 in Chapter 6 of Friedman [12]. These theorems are rewritten for solving the Poisson equation as follows.

Theorem 1
Define a function $\Phi(\mathbf{x})$ as

$$\Phi(\mathbf{x}) = \mathrm{E}^{\mathbf{x}}\left[G(\mathbf{x}(\tau))\right] - \mathrm{E}^{\mathbf{x}}\left[\int_0^\tau \mathrm{d}t\, F(\mathbf{x}(t))\right], \tag{2}$$

where $\mathrm{E}^{\mathbf{x}}$ denotes the expectation with respect to the probability law of sample paths $\{\mathbf{x}(t); 0 \le t\}$, which start from a position $\mathbf{x}$ at time $t = 0$, i.e., $\mathbf{x}(0) = \mathbf{x}$. Here, the stochastic process $\mathbf{x}(t)$, which describes a path of random walking of a point particle, is given by

$$\mathrm{d}\mathbf{x}(t) = \sqrt{2}\,\mathrm{d}\mathbf{W}(t) \tag{3}$$

in Cartesian coordinates $(x^1, x^2, x^3) = (x, y, z)$, where $\mathbf{W}(t)$ is a Wiener process, and τ is the first exit time of the stochastic process $\mathbf{x}(t)$ from the domain of the function Φ. Then $\Phi(\mathbf{x})$ is the unique solution of the Poisson equation:

$$\nabla^2\Phi = F(\mathbf{x}), \tag{4a}$$

where the boundary condition is defined at the boundary of the domain as follows:

$$\Phi(\mathbf{x}) = G(\mathbf{x}). \tag{4b}$$

Here, F and G are arbitrary functions of $\mathbf{x}$. It should be noted that the physical quantities shown in this theorem are normalized in SI units using, for example, $x_0 = 1$ [m], $m_0 = 1$ [kg], $t_0 = 1$ [s], and $e_0 = 1$ [C].

Equation (2) represents the stochastic solution of the Poisson equation given by the theory of stochastic differential equations. The contributions of the point particle to the solution are described as the value of G at the point $\mathbf{x}(\tau)$ in the first term and the time integral along the path $\mathbf{x}(t)$ from time $t = 0$ to τ in the second term. Using this solution, we can calculate a value for the electrostatic potential, Φ, at an arbitrary point $\mathbf{x}$ in the domain. Notably, Theorem 1 is not directly applied to the Poisson equation in curvilinear coordinates; thus, Eq. (3) requires modification. We will discuss more detail on this point later.

The Poisson solver is first applied to the following two-dimensional (2D) Poisson problem, to provide a benchmark for it. In the Poisson Eq. (4a) in the Cartesian space, the charge density ρ is set to $\rho = \rho(x, y) = 1.0 \times 10^{-8}$ [C/m^3] if $r = \sqrt{x^2 + y^2} < r_0$ and $\rho = 0$ otherwise, where r_0 is set to $r_0/a = 0.25$ and $r(x, y) = a = 1$ [m] is the boundary at which $\Phi = G = 0$. The analytic solution is presented in the work by Endo [13]:

$$\Phi(r) = \begin{cases} \dfrac{\rho r_0^2}{2\epsilon_0}\left[\ln\left(\dfrac{a}{r_0}\right) + \dfrac{1}{2} - \dfrac{r^2}{2r_0^2}\right] & \text{if } 0 \le r \le r_0, \\ \dfrac{\rho r_0^2}{2\epsilon_0}\ln\left(\dfrac{a}{r}\right) & \text{if } r_0 < r \le a. \end{cases} \tag{5}$$

This 2D analytic solution is derived under the assumption of cylindrical symmetry. The numerical solution given by the Poisson solver is confirmed to be consistent with the analytic solution (5), as presented in Fig. 1. There is a numerical error caused by a difference scheme for solving the

stochastic differential Eq. (3), which is numerically solved using the second order Runge-Kutta method [14, 15] with a time step Δt, where Δt is a fixed value in the solver and is set to $\Delta t/t_0 = 1.0 \times 10^{-4}$ in Fig. 1. Here, the normalizing time scale, t_0, and length, x_0, are $t_0 = 1$ [s] and $x_0 = a = 1$ [m], respectively, which are estimated from the diffusion coefficient 2 [m^2/s] given by Eq. (3). When the time step is increased to, for example, 25 times that of Fig. 1, the numerical solution degrades, as shown in Fig. 2 (a). The numerical error evaluated by $|\Phi^{\rm num} - \Phi^{\rm ana}|/|\Phi^{\rm ana}|$ is shown in Fig. 2 (b), where $\Phi^{\rm num}$ is the numerical solution and $\Phi^{\rm ana}$ is the analytic solution. The numerical error becomes larger, particularly around the boundary $r/a \sim 1$. This is because the numerical accuracy of the time integral along the sample paths given by Eq. (3) degrades if the starting point of the paths is close to the boundary. Notably, the second term in the solution (2) is given by the time integral along the sample paths, and the first term is zero in this case. In contrast to the cases of $r/a \sim 1$, the numerical error is smaller when the starting point is sufficiently far from the boundary. For instance, the error is $\lesssim 10^{-2}$ for $|x/a| \lesssim 0.3$ in Fig. 1. The following should be noted: the aforementioned results imply that the spatial resolution of this Poisson solver depends on the time step Δt, i.e., the step sizes of the random walkers, with the root mean square of the step sizes being $\sim \sqrt{2\Delta t}$. The spatial resolution will be discussed again later. The degree of numerical error shown in this problem should also depend on the total number of random walkers per point. In Figs. 1 and 2, the total number, $N_{\rm rw}$, is set to 72,000 at each point $\mathbf{x}$ in the domain. The error is proportional to $\sim 1/\sqrt{N_{\rm rw}}$. Figure 3 shows that the error becomes approximately 10 times greater when $N_{\rm rw}$ is adjusted to $N_{\rm rw}/100$.

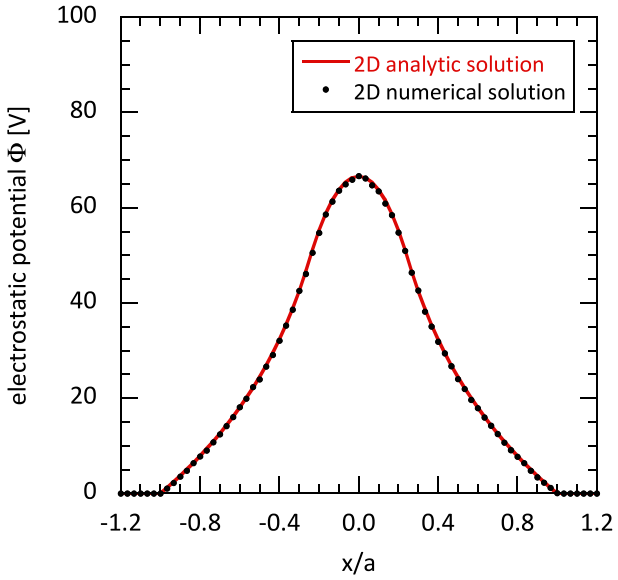

Fig. 1 The black dots represent the numerical solution given by the Poisson solver. The red solid line depicts the analytic solution given in Eq. (5). Both the solutions are calculated along the x axis.

Next, we applied the solver to a 2D example that employs the kanji characters '春夏秋冬' which produce a complicated boundary with the conditions $F(\mathbf{x}) = 1$ and $G(\mathbf{x}) = 0$ for verifying the potential calculation in a specified part of the domain. First, when applying the solver to the entire domain, the potential, $\Phi(\mathbf{x})$, is shown in Fig. 4. Maintaining the conditions of F and G in Fig. 4, this solver can also be used to calculate the potential $\Phi(\mathbf{x})$ in the specified part, as can be seen from Fig. 5. It should be noted that the paths $\mathbf{x}(t)$ of the random walkers starting from a point

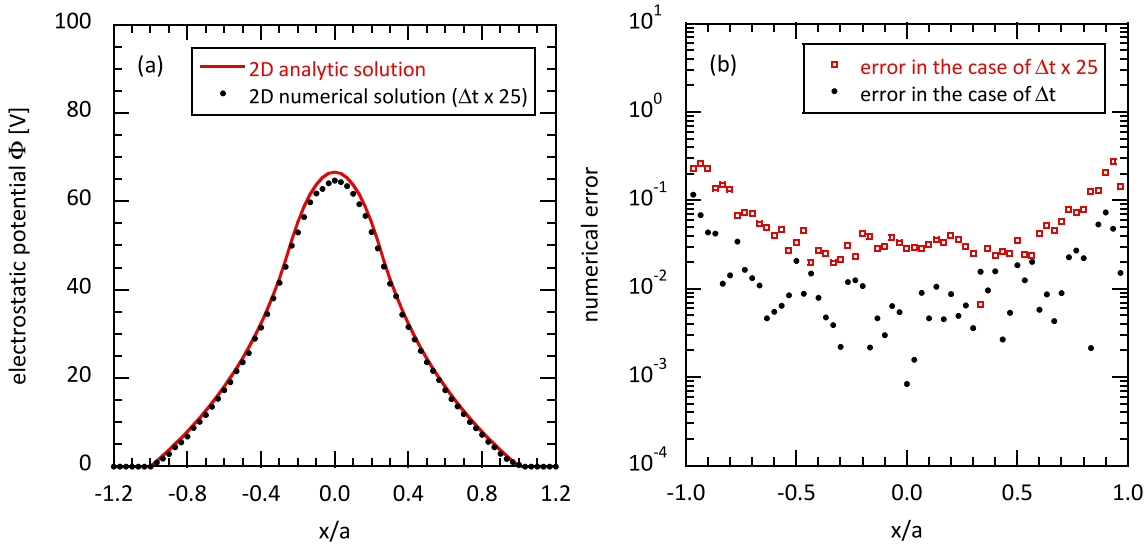

Fig. 2 (a) The numerical solution in the case where $\Delta t/t_0$ is 25 times greater than in Fig. 1 is represented by the black dots. The analytic solution is illustrated by the red solid line. Both solutions are calculated along the x axis. (b) The numerical errors are shown by the black dots for the case of Fig. 1 and by the red open squares for the case of Fig. 2 (a).

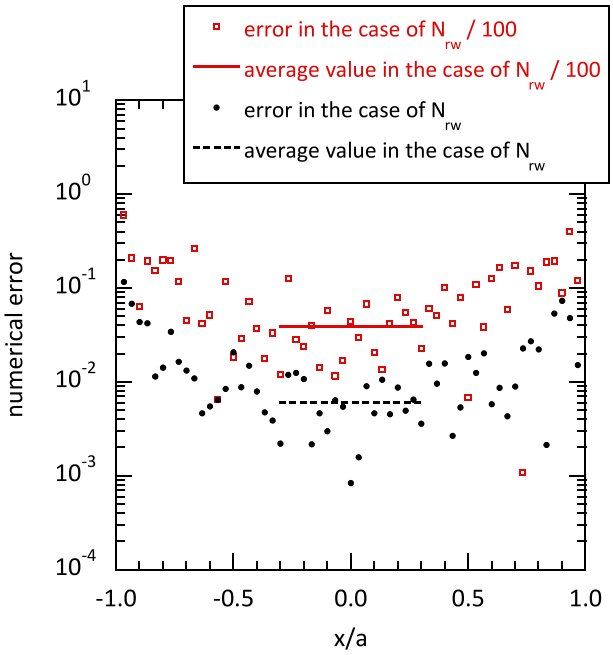

Fig. 3 The numerical errors are plotted by the black dots for the case of $N_{\rm rw} = 72,000$ per point, which is the case in Fig. 1, and represented by the red open squares for the case of $N_{\rm rw}/100 = 720$ per point. The average values of the error for $|x/a| \leq 0.3$ are shown by the black dashed line for the case of $N_{\rm rw}$ and by the red solid line for the case of $N_{\rm rw}/100$.

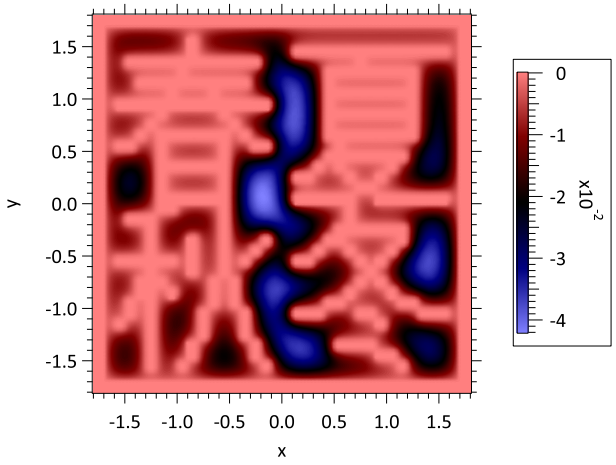

Fig. 4 The numerical solution given by the Poisson solver for the case of $G(\mathbf{x}) = 0$ in the region in red, which is given by the KANJI characters '春夏秋冬', and $F(\mathbf{x}) = 1$ in the domain.

x are kept to be followed even after they have gone outside the specified part and until they firstly cross the boundary given by G, i.e., until the time $t = \tau$. Here, the first exit time τ is determined probabilistically for each random walker by firstly crossing the boundary. It should be also noted that the probability used for calculating the expectation in Eq. (2) is conserved as its total since the random

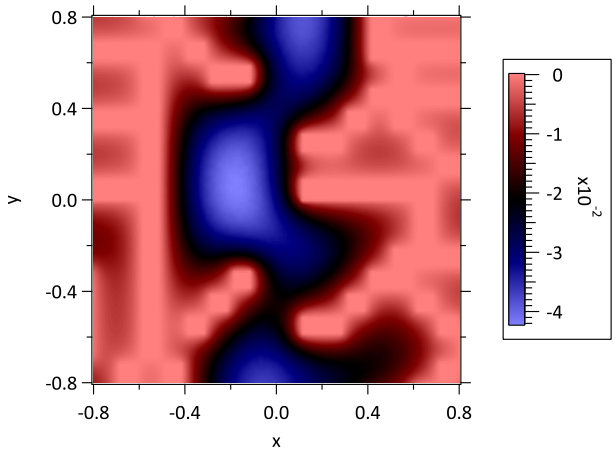

Fig. 5 The numerical solution given by the Poisson solver for the case of calculating the potential, Φ, only in the region of $|x| \leq 0.8$ and $|y| \leq 0.8$ in Fig. 4.

walkers do not disappear until they firstly cross the boundary. Here, the probability is given by the random walkers in the MC method. As illustrated in Fig. 6 (a), there is no degradation in the accuracy of the numerical solution when the potential is calculated only in the specified part. If a solver based on finite-difference or finite-element methods is used to calculate the potential in Fig. 5, the values of G at the boundary should be reset, where the boundary is defined by the parts of the kanji characters and the edge of the image itself in Fig. 5.

The potential, Φ, (and its gradient) at a point **x** depends on the spatial structures of F and G, which the random walkers starting from the point **x** can access until they firstly cross the boundary given by G. As shown in Fig. 6 (a), the gradients of Φ at, for example, $x = -0.5$ and $x = 0.7$ are different depending on the spatial structure around these points. If the spatial resolution of the method proposed in this study degrades, i.e., when the time step, Δt, is modified to be, for example, 25 times greater than that in Fig. 6 (a), the gradients at $x = -0.5$ and $x = 0.7$ become almost identical, as shown by the black solid line in Fig. 6 (b). This is because the random walkers do not sufficiently detect the spatial structures of F and G. The potential, Φ, converges to the values shown in Fig. 6 (a) when the spatial resolution improves. See, for example, the green dashed line and the red solid line in Fig. 6 (b).

The stochastic process, $\mathbf{x}(t)$, depends on the coordinate system in which the Poisson equation is solved, as described in the study by Itô [9]. Hereafter, 3D curvilinear coordinates (u^1, u^2, u^3) are used in solving the Poisson equation:

$$F(u^1, u^2, u^3) = \nabla^2 \Phi(u^1, u^2, u^3) = \frac{1}{\sqrt{\mathrm{g}}} \frac{\partial}{\partial u^k} \left\{ \sqrt{\mathrm{g}}\, \mathrm{g}^{k\ell} \frac{\partial}{\partial u^\ell} \Phi \right\}$$

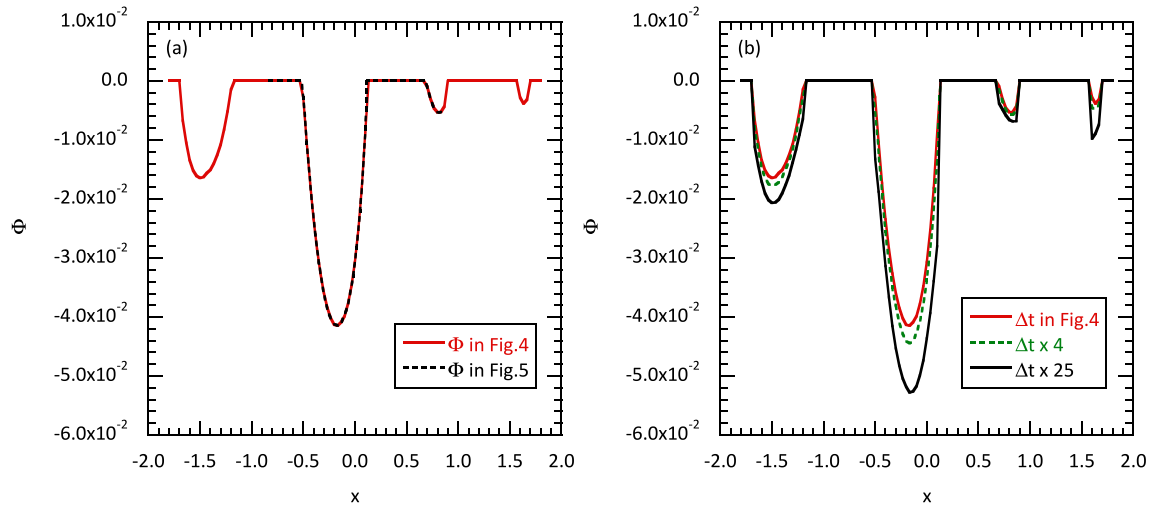

Fig. 6 The numerical solution, Φ, is shown with respect to x under the condition of $y = 0$ in (a) by the red solid line for the case in Fig. 4 and the black dashed line for the case in Fig. 5. In (b), the numerical solution, Φ, is also shown by the red solid line for the case of $\Delta t/t_0 = 1.0 \times 10^{-4}$, which is also set in (a), the green dashed line for the case of $\Delta t/t_0 = 4.0 \times 10^{-4}$, and the black solid line for the case of $\Delta t/t_0 = 2.5 \times 10^{-3}$.

$$= U^k \frac{\partial}{\partial u^k} \Phi + \frac{1}{2} D^{k\ell} \frac{\partial^2}{\partial u^k \partial u^\ell} \Phi, \tag{6a}$$

and

$$\Phi(u^1, u^2, u^3) = G(u^1, u^2, u^3) \tag{6b}$$

at the boundary of the domain. The stochastic solution of Eqs. (6a) and (6b) is the same as Eq. (2), but the stochastic process $\mathbf{x}(t) = (u^1(t), u^2(t), u^3(t))$ is modified as follows:

$$\mathrm{d}u^k(t) = U^k \mathrm{d}t + \sigma^{k\ell} \mathrm{d}W_\ell(t). \tag{7}$$

Here, $D^{k\ell} = 2\mathfrak{g}^{k\ell}$ is the diffusion coefficient tensor, $\mathfrak{g}^{k\ell}$ is the metric of the curvilinear coordinates, $\mathfrak{g} = \det[\mathfrak{g}_{k\ell}]$, and U^k and $\sigma^{k\ell}$ are given as

$$U^k = \frac{1}{\sqrt{\mathfrak{g}}} \frac{\partial}{\partial u^j} \left\{ \sqrt{\mathfrak{g}}\, \mathfrak{g}^{jk} \right\}, \tag{8a}$$

and

$$\sigma^{k\ell} = \sqrt{2}\, \mathfrak{g}^{kj} \left\{ \frac{\partial x^\ell}{\partial u^j} \right\}, \tag{8b}$$

where (x^1, x^2, x^3) is the Cartesian coordinate system. From Eq. (7), the relation between D^{jk} and $\sigma^{\alpha\beta}$ is confirmed as follows:

$$\begin{aligned} D^{jk} &= \lim_{\varepsilon \to 0+} \frac{1}{\varepsilon} \mathrm{E}\left[\left\{ u^j(t+\varepsilon) - u^j(t) \right\} \left\{ u^k(t+\varepsilon) - u^k(t) \right\} \right] \\ &= \lim_{\varepsilon \to 0+} \frac{1}{\varepsilon} \mathrm{E}\left[\sigma^{j\alpha} \left\{ W_\alpha(t+\varepsilon) - W_\alpha(t) \right\} \right. \\ &\qquad \left. \times \sigma^{k\beta} \left\{ W_\beta(t+\varepsilon) - W_\beta(t) \right\} \right] \\ &= \sigma^{j\alpha} \sigma^{k\beta} \delta_{\alpha\beta} = 2\, \mathfrak{g}^{jk}, \end{aligned} \tag{9}$$

where E denotes the expectation with respect to the probability measure of $\mathbf{x}(t)$ and the Wiener process, $\mathbf{W}(t) = (W_1(t), W_2(t), W_3(t))$, satisfies the following for arbitrary time $t \geq 0$ [11, 12]:

$$\mathrm{E}\left[\left\{ W_\alpha(t+\varepsilon) - W_\alpha(t) \right\} \left\{ W_\beta(t+\varepsilon) - W_\beta(t) \right\} \right] = \varepsilon\, \delta_{\alpha\beta}. \tag{10}$$

We use the curvilinear coordinates (u^1, u^2, u^3) shown below to benchmark the solver, which is a special case of the helical coordinates [16],

$$x = \left(R_{\mathrm{ax}} + u^1 \right) \cos u^3, \tag{11a}$$

$$y = -\left(R_{\mathrm{ax}} + u^1 \right) \sin u^3, \tag{11b}$$

$$z = u^2. \tag{11c}$$

The calculation of the potential, Φ, is addressed when the charge density ρ is set to $\rho = \rho(u^1, u^2, u^3) = 1.0 \times 10^{-8}$ [C/m^3] if $r = \sqrt{(u^1)^2 + (u^2)^2} < r_0$ and $\rho = 0$ otherwise, where r_0 is set to $r_0/a = 0.25$ and $r(u^1, u^2) = a = 1$ [m] is the boundary at which $\Phi = G = 0$. Here, R_{ax} is the major radius of the axis of the torus and the coordinates (u^1, u^2, u^3) satisfy $-R_{\mathrm{ax}} < u^1, u^2 < R_{\mathrm{ax}}$ and $0 \leq u^3 < 2\pi$. In this case, U^k in the Poisson Eq. (6a) is given from Eq. (8a) as follows:

$$U^1 = \frac{1}{R_{\mathrm{ax}} + u^1}, \tag{12a}$$

$$U^2 = U^3 = 0. \tag{12b}$$

We also obtain $\sigma^{k\ell}$ from Eq. (8b) as follows:

$$\sigma^{11} = \sqrt{2} \cos u^3, \tag{13a}$$

$$\sigma^{12} = -\sqrt{2} \sin u^3, \tag{13b}$$

$$\sigma^{23} = \sqrt{2}, \tag{13c}$$

$$\sigma^{31} = -\frac{\sqrt{2}}{R_{\mathrm{ax}} + u^1} \sin u^3, \tag{13d}$$

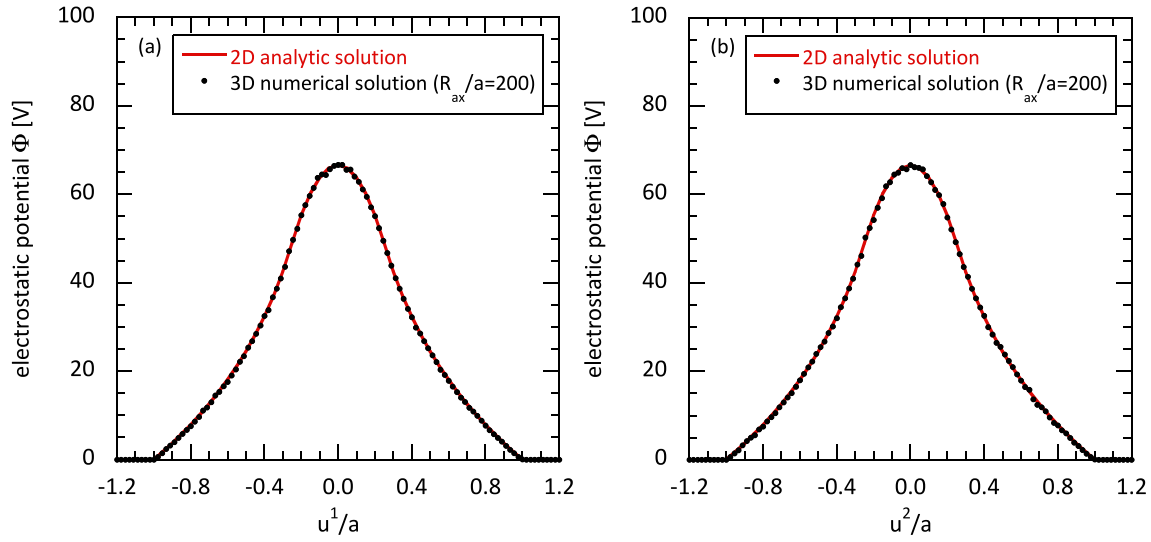

Fig. 7 The numerical solution given by the Poisson solver in the case of $R_{\rm ax}/a = 200$ is represented by the black dots. The 2D analytic solution given in Eq. (5) is illustrated by the red solid lines. Both solutions are calculated along (a) the u^1 axis and (b) the u^2 axis.

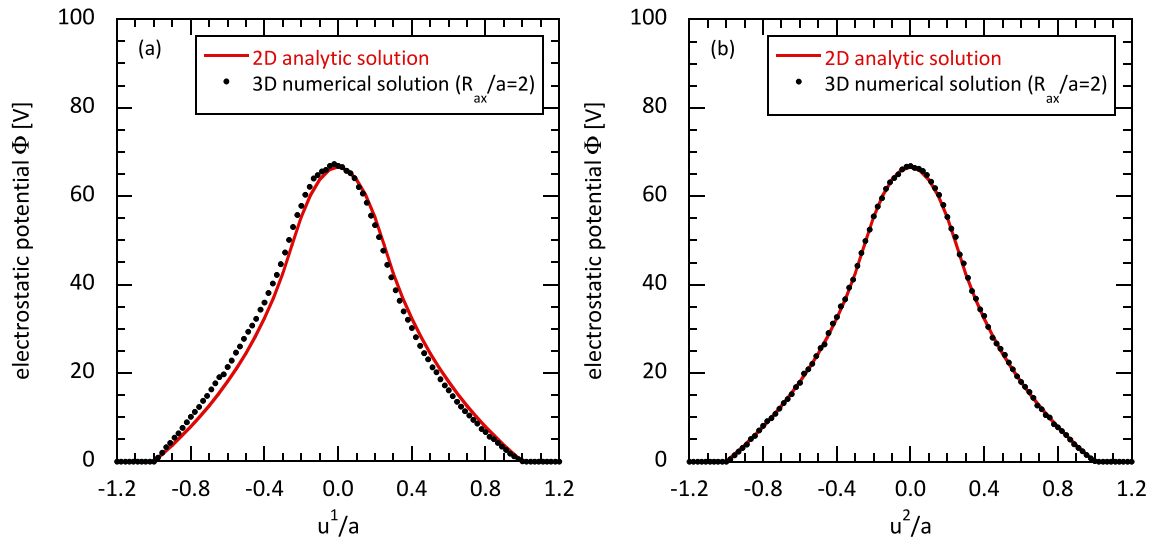

Fig. 8 The numerical solution given by the Poisson solver in the case of $R_{\rm ax}/a = 2$ is shown by the black dots. The 2D analytic solution given in Eq. (5) is illustrated by the red solid lines. Both solutions are calculated along (a) the u^1 axis and (b) the u^2 axis.

$$\sigma^{32} = -\frac{\sqrt{2}}{R_{\rm ax} + u^1} \cos u^3, \tag{13e}$$

$$\sigma^{13} = \sigma^{21} = \sigma^{22} = \sigma^{33} = 0. \tag{13f}$$

From Eq. (9), it is confirmed that the tensor $\sigma^{k\ell}$ is related to the diffusion coefficient tensor $D^{k\ell}$, where $D^{11} = D^{22} = 2$, $D^{33} = 2/(R_{\rm ax} + u^1)^2$, and $D^{k\ell} = 0$ if $k \neq \ell$. The relationship $D^{k\ell} = 2\mathfrak{g}^{k\ell}$ is also confirmed in the coordinates (u^1, u^2, u^3) that satisfy Eqs. (11a) - (11c). The difference between the 3D numerical solution evaluated by the solver and the 2D analytic solution given in Eq. (5) becomes smaller if the torus is slender, i.e., $U^1, D^{33} \to 0$ as $R_{\rm ax}/a \to \infty$, $U^2 = U^3 = 0$, $D^{k\ell} = 0$ for $k \neq \ell$ and $D^{11} = D^{22} = 2$ in Eq. (6a). This tendency is confirmed in Fig. 7. The 3D numerical solution is almost identical to the 2D analytic solution, when the major radius is set to $R_{\rm ax}/a = 200$. Here, the degree of numerical error depends on both the time step and the total number of random walkers, as discussed in Figs. 2 and 3. If the major radius is reduced, for example $R_{\rm ax}/a = 2$, the 3D numerical solution differs from the 2D analytic solution (Fig. 8). Figure 8 (a) shows that due to toroidicity, the 3D numerical solution on the u^1-axis is slightly larger in the region of $-1 < u^1/a < 0$ and slightly smaller in the region of $1 > u^1/a > 0$ compared with the 2D analytic solution given in Eq. (5). Conversely, the 3D numerical solution on the u^2-axis is consistent with the 2D analytic solution and satisfies the up-down symmetry,

as presented in Fig. 8 (b). For checking dependence of the numerical solution on random numbers used in the solver, the difference between the solutions calculated multiple times should be evaluated using different random number sequences [10]. In this study, the difference is given as $\int_{\mathcal{D}} \mathrm{d}^3x \left|\Phi^{(1)}(\mathbf{x}) - \Phi^{(2)}(\mathbf{x})\right| / \int_{\mathcal{D}} \mathrm{d}^3x\,(1/2)\left|\Phi^{(1)}(\mathbf{x}) + \Phi^{(2)}(\mathbf{x})\right|$, where $\Phi^{(1)}$ and $\Phi^{(2)}$ are the numerical solutions calculated twice and the difference is evaluated in the domain $\mathcal{D}$. In the case of Fig. 8, this difference is 1.9×10^{-2}.

3. Summary and Discussion

In this study, the Monte Carlo Poisson solver is developed, based on the theory of stochastic differential equations, for partly calculating electrostatic potential, Φ, in 3D curvilinear coordinates. The computational cost of calculating the potential, Φ, derived from the Poisson equation becomes problematic, when impurity transport is considered in the edge region ergodized by the RMPs. Here, the cost includes the preparations for the potential calculation, as discussed in the introduction. The computational cost may increase because the magnetic structure, which significantly affects the charge density, is complicated by the RMPs. Further, if traditional techniques (such as finite-difference or finite-element methods) are used for evaluating the potential, Φ, when the boundary condition is set on the vacuum vessel, Φ is calculated throughout the entire domain within the vacuum vessel. We demonstrate that the potential, Φ, can be successfully calculated only in a selected region within the entire domain using the stochastic solution of the Poisson equation shown in Theorem 1. Thus, the MC solver can reduce the cost by approximately [the volume of the edge region] / [the total volume within the vacuum vessel] in 3D curvilinear coordinates without requiring advanced techniques for the mesh system and setting the boundary condition in the selected region. It should be noted that Theorem 1 is not directly applied to the Poisson equation in 3D curvilinear coordinates, and that this application particularly requires Eq. (8b), which is derived in this study.

The numerical accuracy of the MC solver's benchmarks is $\lesssim 10^{-2}$. This accuracy is sufficient for estimating impurity particle flux in one or two significant digits in a kinetic simulation of impurity transport in a quasi-steady state. If more accurate calculation of the potential is required, the total number of random walkers should be increased and their time step should be finer. The boundary propagation method is useful in this scenario for decreasing the MC solver's computational cost [17], where the boundary propagation is illustrated in Fig. 9. To improve the numerical accuracy and reduce the computational cost of the potential calculation, combining the MC method and the finite-difference or finite-element methods is also beneficial. When a solver based on the finite-difference or finite-element methods is applied only to a selected region within the entire domain, the boundary condition of that region is provided by the MC solver proposed in this study.

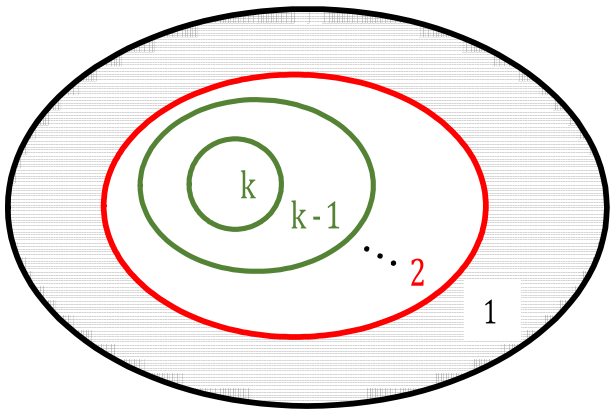

Fig. 9 Illustration of the boundary propagation [17]. The original boundary is shown by the black solid-line with number 1. The 2nd boundary is shown by the red solid-line. The region between the original boundary (i.e., the 1st boundary) and the 2nd boundary is hatched in this figure. The procedure consists of initiating random walks from points within the region between the $(k-1)$-th and the k-th boundaries, where $k = 2, 3, 4, \cdots$. Here, the boundary condition in calculating the potential within this region is set at the $(k-1)$-th boundary. The $(k-1)$-th boundary is replaced by the k-th boundary when all random walks have terminated.

In general, the electrostatic potential in and around the edge region depends on several physics mechanisms, including neoclassical and turbulent transport and interactions between various particles such as background ions, impurities, electrons and neutrals. However, these mechanisms are not discussed here. Simulation studies of impurity transport will be conducted in the future using the MC solver developed in this study.

Acknowledgments

The computational simulations in this paper were conducted mainly using the supercomputer 'Plasma Simulator' at National Institute for Fusion Science (NIFS). This work was supported in part by the NIFS Collaborative Research Programs NIFS19KNTT052, NIFS21KNTT060 and NIFS20KNST164, the IFERC-CSC project DKSOIT, and JSPS KAKENHI Grant Number JP20K03908.

[1] T. Evans *et al.*, Phys. Rev. Lett. **92**, 235003 (2004).
[2] B.S. Victor *et al.*, Plasma Phys. Control. Fusion **62**, 095021 (2020).
[3] S. Mordijck *et al.*, Nucl. Fusion **54**, 082003 (2014).
[4] O. Schmitz *et al.*, Plasma Phys. Control. Fusion **50**, 124029 (2008).
[5] W.H. Press *et al.*, *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 2007).
[6] F. Zhang *et al.*, Eng. Comput. **32**, 285 (2006).
[7] T. Moritaka *et al.*, Plasma **2**, 179 (2019).

[8] J.M. DeLaurentis and L.A. Romero, J. Comput. Phys. **90**, 123 (1990).
[9] K. Itô, *The Brownian motion and tensor fields on Riemannian manifold*, in Proceedings of the International Congress of Mathematicians (International Congress of Mathematicians, 15 - 22 August 1962, Stockholm), p.536-539.
[10] R. Kanno *et al.*, Plasma Fusion Res. **6**, 2403066 (2011).
[11] B. Øksendal, *Stochastic Differential Equations* (Springer-Verlag, Berlin Heidelberg, 2003).
[12] A. Friedman, *Stochastic Differential Equations and Applications* (Dover, New York, 2004).
[13] M. Endo, 世界一易しい*Poisson*方程式シミュレーション, http://teamcoil.sp.u-tokai.ac.jp/lectures/EM1/Poisson/index.html (in Japanese).
[14] J.R. Klauder and W.P. Petersen, SIAM J. Numer. Anal. **22**, 1153 (1985).
[15] T. Misawa and H. Itakura, Phys. Rev. E **51**, 254 (1995).
[16] K. Harafuji *et al.*, J. Comput. Phys. **81**, 169 (1989).
[17] R.E. Hiromoto and R.G. Brickner, *Empirical Results of a Hybrid Monte Carlo Method for the Solution of Poisson's Equation*, In: Brebbia C.A., Peters A., Howard D. (eds) Applications of Supercomputers in Engineering II. (Springer, Dordrecht, 1991) p.233-239.