CONFERENCE PRE-PRINT

DEVELOPMENT OF A THREE-DIMENSIONAL SIMULATION CODE FOR SCRAPE-OFF LAYER PLASMAS

J.F. HE
University of Science and Technology of China
Hefei, China
Email: hejiaf@mail.ustc.edu.cn

S.F. MAO
University of Science and Technology of China
Hefei, China
Email: sfmao@ustc.edu.cn

M.Y. YE
University of Science and Technology of China
Hefei, China
Email: yemy@ustc.edu.cn

Abstract

A three-dimensional (3D) scrape-off layer (SOL) plasma code based on the finite volume method has been preliminarily developed. According to the 3D fluid equations of continuity, parallel momentum, current and ion and electron internal energy, which are derived in a similar way of SOLPS-ITER equations without the restriction of toroidal symmetry, the 3D SOL code is developed using C++ based on the finite volume method. The numerical calculations are performed on the collocated grid, where the convection term is discretized using a hybrid scheme, the diffusion term is discretized using a central difference scheme, and the time differential term is discretized using a first-order fully implicit scheme. Based on the example ASDEX-Upgrade case of SOLPS-ITER for the B2.5 standalone simulation without currents and drifts, the 3D SOL code is tested. The 3D computational grid is obtained by toroidally extending the 2D grid and uniformly dividing in the toroidal direction. The results are computed for different number of toroidal cells. While the distribution of density, parallel velocity, ion temperature and electron temperature in the poloidal cross section are identical toroidally, the 3D SOL code can well reproduce the 2D profiles. Further test is implemented by introducing a particle source in the SOL at the outer mid-plane and a certain toroidal position. The plasma density increases along the magnetic field, which correctly reflects the effect due to the toroidal asymmetric source.

1. INTRODUCTION

It is critical to control the heat load onto the divertor target for the future fusion reactor. While the introducing of three-dimensional (3D) magnetic perturbations such as resonant magnetic perturbation (RMP) in a tokamak device is proved to be an effective way to control the transient heat load due to the burst of edge-localized modes (ELMs) [1], the distribution of the inter-ELM divertor heat load will be significantly affected simultaneously [2]. Many devices have equipped with RMP coils, such as EAST, HL-2A, JET, DIII-D, ASDEX Upgrade. Theoretical simulation and experiment results have shown that the imposed external RMP field could modify the topology of edge magnetic field and induce the 2D pattern heat flux on divertor target often appearing as strike point splitting. To understand the effect 3D fields on the transport of scrape-off layer (SOL) plasma and hence the divertor heat load, the 3D code EMC3-EIRENE [3] based on Monte Carlo method is widely used. However, it is difficult to consider the effect of currents and drifts due to the restriction of the Monte Carlo method in EMC3-EIRENE code, which have pronounced effects on the divertor condition. On the other hand, pronounced effects on the SOL plasma have been found due to the drifts, which can be well simulated in the 2D code SOLPS-ITER [4] based on the finite volume method. However, it is still lack of an effective way to understand the drift effects on the SOL plasma under 3D field perturbation in a tokamak. One possible solution is to find a reasonable way to introduce the 3D field effect into a fluid code based on finite volume method. In this work, a 3D SOL code is developed for the main ion based on finite-volume method.

2. THE THREE-DIMENSIONAL FLUID EQUATIONS

The three-dimensional fluid equations are derived in a similar way of SOLPS-ITER equations without the restriction of toroidal symmetry under a toroidal geometry with the *x* and *y* coordinates corresponding to the directions along and across the flux surfaces, respectively, while *z* is the toroidal direction, as shown in Fig 1. The metric coefficients are $h_x = \frac{1}{\|\nabla x\|}$, $h_y = \frac{1}{\|\nabla y\|}$, $h_z = \frac{1}{\|\nabla z\|}$, $\sqrt{g} = h_x h_y h_z$. b_x and b_z are the pitches of the magnetic field, B_x is the poloidal component of the magnetic field and B_z is the toroidal component of the magnetic field, $b_x = B_x / B$, $b_z = B_z / B$. The subscript '$\|$' denotes the direction parallel to the magnetic field B.

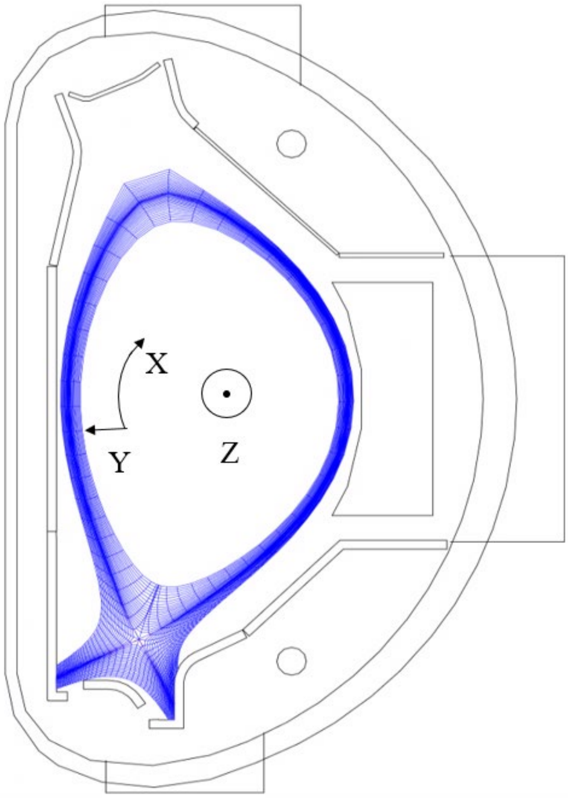

Fig 1. Coordinate system and simulation mesh: x is the poloidal coordinate, y is the radial coordinate orthogonal to the flux surfaces, z is the toroidal coordinate orthogonal to both the x coordinate and y coordinate.

A quasi-neutral plasma and a single ion species are considered with charge state z_i so that density $n_i = n_e / z_i$. The continuity equation for particle conservation (equation (1)), the parallel momentum conservation equation (equation (2)), the ion and electron energy equations(equation (3) and (4)) and the current equation(equation (5)) are shown as follows:

$$\frac{\partial n_a}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}\Gamma_{ax}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}\Gamma_{ay}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{\sqrt{g}}{h_z}\Gamma_{az}\right) = S_a^n \tag{1}$$

$$\begin{aligned} m_a \frac{\partial n_a V_{\|a}}{\partial t} &+ \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{h_z\sqrt{g}}{h_x}\Gamma_{ax}^m\right) + \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{h_z\sqrt{g}}{h_y}\Gamma_{ay}^m\right) + \frac{1}{h_z\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{h_z\sqrt{g}}{h_z}\Gamma_{az}^m\right) + \\ \frac{b_x}{h_x}\frac{\partial n_a T_i}{\partial x} &+ Z_a e n_a \frac{b_x}{h_x}\frac{\partial \phi}{\partial x} + \frac{b_z}{h_z}\frac{\partial n_a T_i}{\partial z} + Z_a e n_a \frac{b_z}{h_z}\frac{\partial \phi}{\partial z} = S_{a\|}^m + S_{CF_a}^m + S_{fr_a}^m + S_{Therm_a}^m + S_a^m \end{aligned} \tag{2}$$

$$
\begin{aligned}
&\frac{3}{2}\frac{\partial n_i T_i}{\partial t}+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}q_{ix}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}q_{iy}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{\sqrt{g}}{h_z}q_{iz}\right)+\\
&\sum_{a=0}^{ns-1}\frac{n_a T_i}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x V_{a\parallel}\right)+\sum_{a=0}^{ns-1}\frac{n_a T_i}{\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{\sqrt{g}}{h_z}b_z V_{a\parallel}\right)=Q_\Delta+Q_{Fab}+\\
&\sum_{fluid\ species}\left(\eta_{ax}\left(\frac{\partial V_{a\parallel}}{h_x\partial x}\right)^2+\eta_a^{(AN)}\left(\frac{\partial V_{a\parallel}}{h_y\partial y}\right)^2+\eta_{az}\left(\frac{\partial V_{a\parallel}}{h_z\partial z}\right)^2\right)+Q_I^{(i)}+Q_R^{(i)}
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
&\frac{3}{2}\frac{\partial n_e T_e}{\partial t}+\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}q_{ex}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}q_{ey}\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{\sqrt{g}}{h_z}q_{ez}\right)\\
&+\frac{n_e T_e}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}b_x V_{e\parallel}\right)+\frac{n_e T_e}{\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{\sqrt{g}}{h_z}b_z V_{e\parallel}\right)=Q_e+Q_{Fei}
\end{aligned}
\tag{4}
$$

$$
\frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\left(\frac{\sqrt{g}}{h_x}j_x\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\left(\frac{\sqrt{g}}{h_y}j_y\right)+\frac{1}{\sqrt{g}}\frac{\partial}{\partial z}\left(\frac{\sqrt{g}}{h_z}j_z\right)=0 \tag{5}
$$

3. CODE STRUCTURE AND NUMERICAL METHOD

3.1. Code structure

The 3D SOL code is developed based on a modular approach and is mainly divided into a data storage module, a discrete transport equation module and a linear matrix solution module, Fig 2. Among them, the data storage module is mainly used to store data such as mesh, plasma states, transport coefficients, source terms, atomic rate coefficients and so on. The equation discretization module is mainly used for discrete transport equations, including the pressure correction module (discrete pressure correction equation), momentum equation module (discrete momentum equation), electron temperature module (discrete electron energy equation), ion temperature module (discrete ion energy equation) and current equation module (discrete current equation). The linear matrix solving module is mainly used to solve the computational matrix constructed by the discrete module.

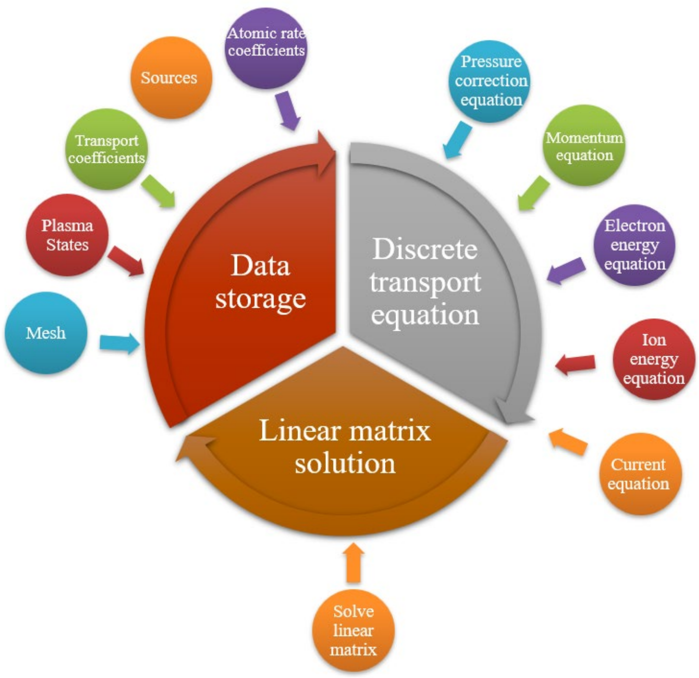

Fig 2.The main structure of the 3D SOL code.

3.2. Discretization method

The 3D SOL code is developed using C++ based on the finite volume method. The numerical calculations are performed on the collocated grid, where the convection term is discretized using a hybrid scheme (equation (6)), the diffusion term is discretized using a central difference scheme (equation (7)), and the time differential term is discretized using a first-order fully implicit scheme (equation (8)).

$$\begin{cases} a_{i+1} = \dfrac{F_{i+\frac{1}{2}}}{2} - \max(D_{i+\frac{1}{2}}, \left|\dfrac{F_{i+\frac{1}{2}}}{2}\right|, \sqrt{D^2_{i+\frac{1}{2}} + \dfrac{F^2_{i+\frac{1}{2}}}{4}}) \\ a_{i-1} = -\dfrac{F_{i-\frac{1}{2}}}{2} - \max(D_{i-\frac{1}{2}}, \left|\dfrac{F_{i-\frac{1}{2}}}{2}\right|, \sqrt{D^2_{i-\frac{1}{2}} + \dfrac{F^2_{i-\frac{1}{2}}}{4}}) \end{cases} \tag{6}$$

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \tag{7}$$

$$\frac{\partial \phi}{\partial t} = \frac{\phi^{t+\Delta t} - \phi^t}{\Delta t} \tag{8}$$

where ϕ denotes a certain physical quantity, the subscript $i+1/2$ denotes the interface between i and $i+1$ control volume, the subscript $i-1/2$ denotes the interface between i and $i-1$ control volume, Δt denotes the time step, the superscript $t+\Delta t$ and t denote the next and current time step, respectively. In the convective term, a is the coefficient of physical quantity ϕ in the control volume, which is determined by its convective flux F and diffusion coefficient D on the control volume interface.

3.3. Computation process

Important advances in Computational Fluid Dynamics (CFD), pertaining to the development and maturity of solution algorithms, have been achieved over last several decades. The density-base algorithm, in which the continuity equation acts as an equation for density while pressure is obtained from the energy and state equations, have been successfully used in the simulation of highly compressible flows. However, density-based algorithms become unstable and their convergence rate greatly diminishes as small disturbances in density result in large variations in the pressure field for low Mach number flows. For the boundary plasma, the typical flow field distribution of SOL is gradually accelerated from a zero-velocity stagnation point around the main plasma to the divertor target plate until the plasma sound speed is reached. Therefore, the density-based algorithm is not suitable for the boundary plasma.

In contrast, the pressure-based algorithm with pressure rather than density as the primitive variable can be used for the numerical calculation of full-velocity fluids. The pressure-based algorithm was first developed as a numerical method for incompressible fluids. While there is no apparent equation governing pressure in incompressible fluids, a pressure or an equivalent pressure-correction equation needs to be derived. The pressure-correction equation can be derived by combining the continuity and momentum equations together in incompressible fluids. Moreover, the pressure-base algorithm can be used for compressible fluids by replacing the density with pressure through the equation of state. The SIMPLE (Semi-Implicit Method for Pressure Linked Equations) [5] is implemented in the 3D SOL code.

The computation process of the code is shown in Fig 3, which includes three main parts: preprocessing, iterative calculation, and postprocessing. Among them, the preprocessing is mainly used to read the input files, including mesh files, particle type files, boundary condition setting files, transport coefficient setting files, particle recycle setting files and plasma initial state setting files, etc. Iterative calculation mainly utilizes the SIMPLE pressure correction algorithm to solve the Braginskii transport equation, which is introduced in Section 2. During this process, the program first calculates the atomic rate coefficients, various transport coefficients and source terms, and then successively solves the momentum equation, pressure correction equation (continuity equation), electron energy equation and ion energy equation until the plasma state converges. The post-processing process mainly outputs the plasma state.

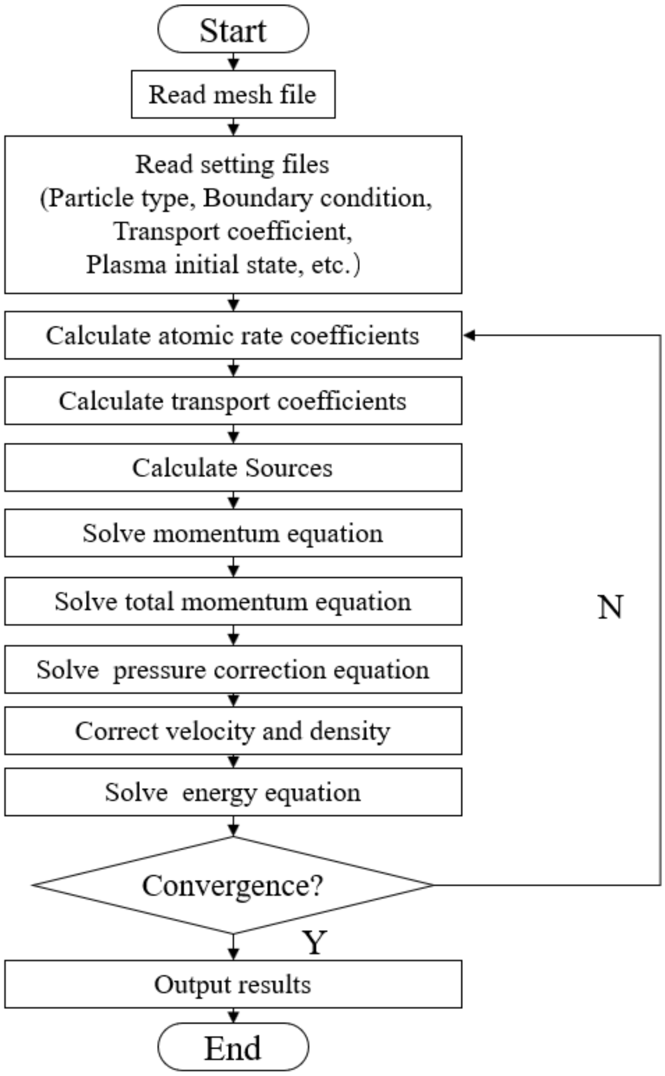

Fig 3. The computation process of the 3D SOL code.

4. SIMULATION RESULT

4.1. Simulation set up

Based on the example ASDEX-Upgrade case of SOLPS-ITER for the B2.5 standalone simulation without currents and drifts, the 3D SOL code is tested. The 3D computational grid is obtained by toroidally extending the 2D grid and uniformly dividing in the toroidal direction, as shown in Fig 4. The boundary condition settings are shown as TABLE 1.

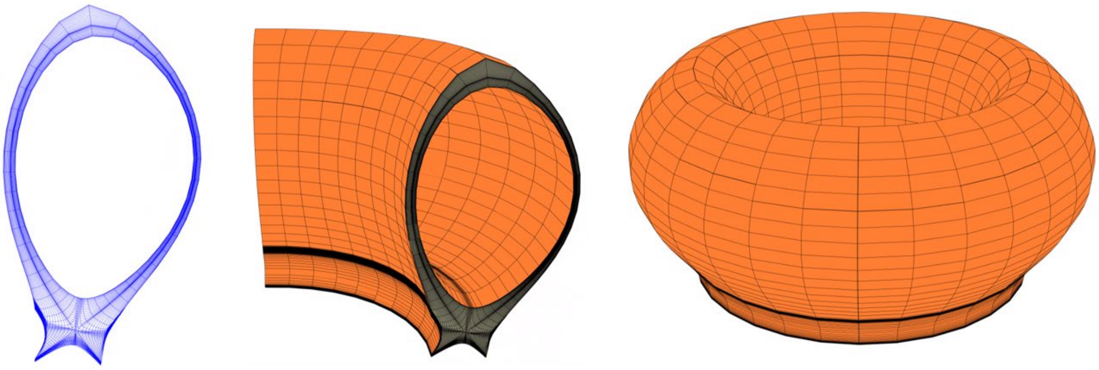

Fig 4. Computational grids for 3D simulation of SOL plasma.

TABLE 1 Boundary condition settings

Equation	Core	Target plate	PFR/SOL
Continuity	Fixed value	Sheath	Leakage
Momentum	Fixed value(mixed)	Sheath	Fixed value(mixed)
Electron energy	Fixed flux	Sheath	Decay
Ion energy	Fixed flux	Sheath	Decay

4.2. Simulation results

For 2D simulations, direct solvers are usually applied since the 2D sparse matrix is not too big, such as the PASTIX solver and SparseLU solver. However, for 3D simulations, the problem takes the form of a 3D sparse matrix to inverse and becomes very costly given that the strong anisotropy of diffusion makes the matrix poorly conditioned. The most efficient solver found to deal with this problem is iterative solver. In this work, the BICGSTAB solver based on iterative stabilized bi-conjugate gradient of Eigen library [6] is used. In order to save CPU time, the parallelization of the computation is implemented relying on OpenMP implementation. The results are computed for different number of toroidal cells. The distribution of density, parallel velocity, ion temperature and electron temperature in the poloidal cross section are identical toroidally. As shown in Fig 5, the 3D SOL code can well reproduce the 2D profiles without any fact of toroidal asymmetry.

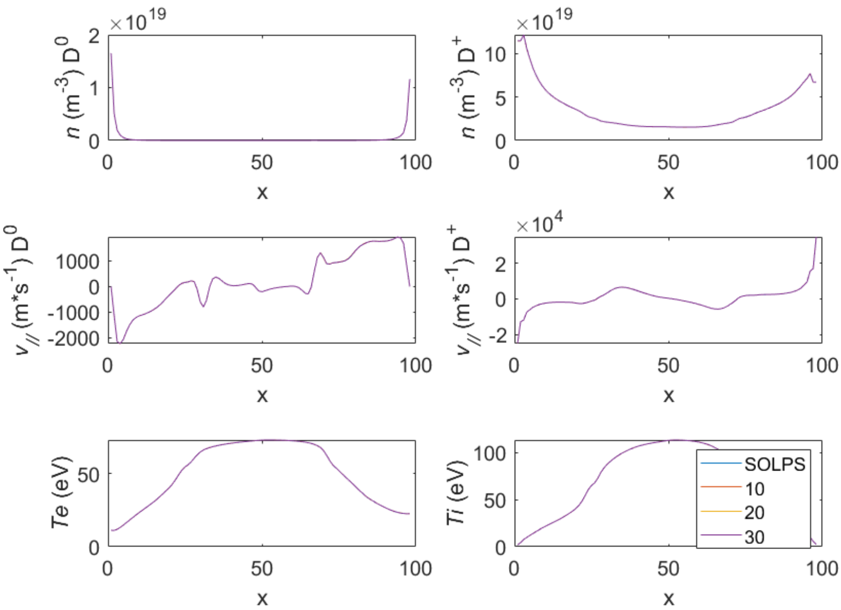

Fig 5. Comparison of the profiles at the separatrix for different number of toroidal cells.

Further test is implemented by introducing a particle source in the SOL at the outer mid-plane and a certain toroidal position. As shown in Fig 6, the plasma density increases along the magnetic field, which correctly reflects the effect due to the toroidal asymmetric source.

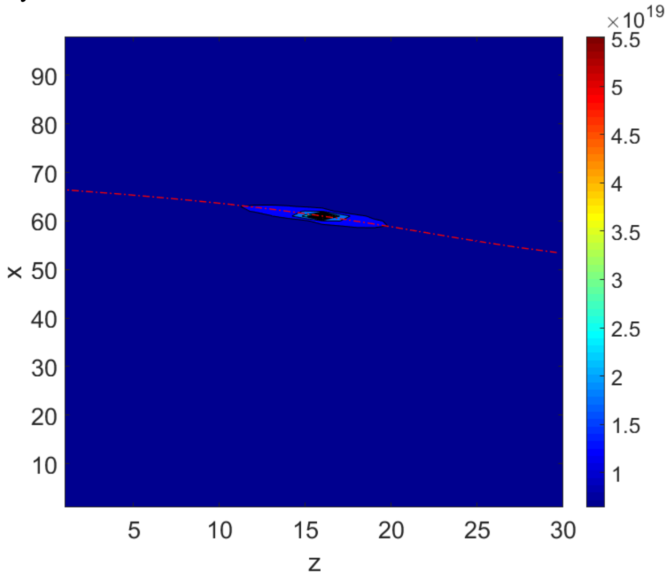

Fig 6. The density distribution on the toroidal (z)-poloidal (x) plane with a particle source of 1×10^{25} $\mathrm{m^{-3}s^{-1}}$ *in the SOL at the outer mid-plane and a certain toroidal position. The red dash line indicates the magnetic field line.*

5. SUMMARY

At the present stage, a 3D scrape-off layer plasma code based on the finite volume method has been preliminarily developed. The 3D SOL code is developed using C++ based on the finite volume method. Without the toroidally asymmetric factor, the 3D SOL code can well reproduce the result for the example ASDEX-Upgrade case of SOLPS-ITER for the B2.5 standalone simulation without currents and drifts. When introducing the assumed particle source in the SOL at a certain toroidal position, the plasma density increases along the magnetic field, which indicates that the effect due to the toroidal asymmetric source can be correctly reflected.

For the purpose to study the SOL plasma with both effects of drifts and 3D fields, further efforts on including the currents and drifts and developing the proper algorithm to introduce 3D fields are ongoing.

ACKNOWLEDGEMENTS

This work was supported by the National MCF Energy R&D Program of China (No. 2019YFE03080500).

REFERENCES

[1] Evans T E, Moyer R A, Burrell K H, et al. Edge stability and transport control with resonant magnetic perturbations in collisionless tokamak plasmas, Nat. Phys. 2 (2006) 419-423.

[2] Ahn J W, Briesemester A R, Kobayashi M, et al. Effect of 3D magnetic perturbations on divertor conditions and detachment in tokamak and stellarator, Plasma Phys. Control. Fusion 59 (2017) 084002.

[3] Feng Y, Sardei F, Kisslinger J, et al. 3D edge modeling and island divertor physics, Contrib. Plasma Phys. 44 (2004) 57–69.

[4] Bonnin X, Dekeyser W, Pitts R, et al. Presentation of the new SOLPS-ITER code package for tokamak plasma edge modelling, Plasma Fusion Res. 11 (2016) 1403102.

[5] Patankar S, Numerical heat transfer and fluid flow, CRC press, (2018).

[6] Eigen, Solving Sparse Linear Systems(2022), https://eigen.tuxfamily.org/dox/group__TopicSparse Systems .html