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

J.F. HE, S.F. MAO, M.Y. YE

University of Science and Technology of China, Hefei, China

Email: sfmao@ustc.edu.cn; yemy@ustc.edu.cn

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 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]. 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. 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, as a preliminary step, a 3D finite-volume code without drifts is developed for simulating the SOL plasma. According to the 3D fluid equations of continuity, parallel momentum and ion and electron internal energy, which are derived in a similar way of B2 equations [5] without the restriction of toroidal symmetry, the 3D 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.

The 3D SOL code is mainly composed of four computational modules, which are momentum module, pressure correction module, electron energy module and ion energy module. For the SOL plasma, the parallel velocity gradually increases from zero at the stagnation point around the main plasma to the sound speed at divertor targets. Therefore, the pressure-based algorithm SIMPLE (Semi-Implicit Method for Pressure Linked Equations) [6] is implemented in the computation rather than the density-based algorithm, which is not suitable for the SOL plasma because it becomes 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. The computation starts with momentum equation, then the pressure correction equation is solved to update particle density and parallel velocity. Finally, the electron and ion temperature are determined by solving the electron and ion energy equations.

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 [7] is used. In order to save CPU time, the parallelization of the computation is implemented relying on OpenMP implementation.

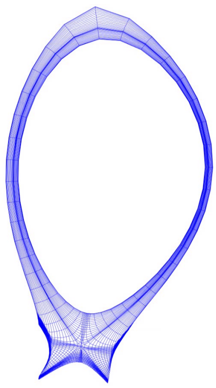

Fig. 1 Computational grid for 3D simulation of SOL plasma.

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. 1. 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 2, the 3D SOL code can well reproduce the 2D profiles without any fact of toroidal asymmetry.

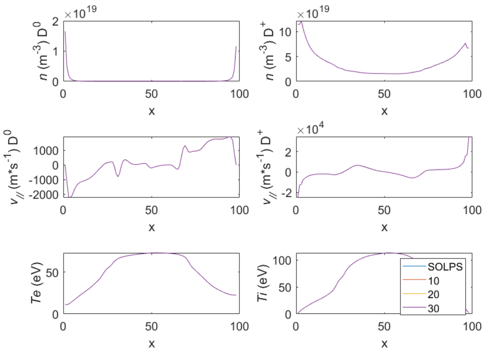

Fig. 2 Comparison of the profiles at the separatrix for different number of toroidal cells.

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. 3, the plasma density increases along the magnetic field, which correctly reflects the effect due to the toroidal asymmetric source.

At the present stage, a 3D scrape-off layer plasma code based on the finite volume method has been preliminarily developed. 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.

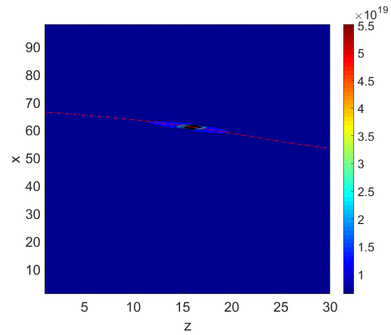

Fig. 3 The density distribution on the toroidal (z)-poloidal (x) plane with a particle source of 1×10^{25} $m^{-3}s^{-1}$ in the SOL at the outer mid-plane and a certain toroidal position. The red dash line indicates the magnetic field line.

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] M. Baelmans, Code improvements and applications of a two-dimensional edge plasma model for toroidal devices, Forschungszentrum Juelich, (1994).

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

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