

REVIEW ARTICLE

Boundary Plasma Turbulence Simulations for Tokamaks[†]

X. Q. Xu^{1,2,*}, M. V. Umansky¹, B. Dudson³ and P. B. Snyder⁴

¹ Lawrence Livermore National Laboratory, L-630, P.O. Box 808, 7000 East Ave., Livermore, CA 94550, USA.

² Institute of Fusion Theory and Simulation, Zhejiang University, Hangzhou 310027, China.

³ Department of Physics, University of York, Heslington, York, YO10 5DD, UK.

⁴ General Atomics, 13-303, P.O. Box 85608, San Diego, CA 92186-5608, USA.

Received 15 May 2008; Accepted (in revised version) 5 June 2008

Available online 3 July 2008

Abstract. The boundary plasma turbulence code BOUT models tokamak boundary-plasma turbulence in a realistic divertor geometry using modified Braginskii equations for plasma vorticity, density (n_i), electron and ion temperature (T_e , T_i) and parallel momenta. The BOUT code solves for the plasma fluid equations in a three dimensional (3D) toroidal segment (or a toroidal wedge), including the region somewhat inside the separatrix and extending into the scrape-off layer; the private flux region is also included. In this paper, a description is given of the sophisticated physical models, innovative numerical algorithms, and modern software design used to simulate edge-plasmas in magnetic fusion energy devices. The BOUT code's unique capabilities and functionality are exemplified via simulations of the impact of plasma density on tokamak edge turbulence and blob dynamics.

PACS: 52.55.Fa, 52.25.Fi, 52.30.EX, 52.30.Gz, 52.35.Ra, 52.65.Tt, 52.65.Kj, 52.65.-y

Key words: Plasma turbulence simulation, plasma two-fluids equation, field-aligned coordinates, plasma blobs.

Contents

1 Introduction	950
----------------	-----

[†]Dedicated to Professor Xiantu He on the occasion of his 70th birthday.

*Corresponding author. *Email addresses:* xxu@llnl.gov (X. Q. Xu), umansky@llnl.gov (M. V. Umansky), bd512@york.ac.uk (B. Dudson), snyder@fusion.gat.com (P. B. Snyder)

2	BOUT dynamical equations	953
3	Magnetic geometry	958
4	Spatial numerical implementations	962
5	Solving BOUT equations with PVODE	966
6	BOUT software design	968
7	BOUT simulation results	969
8	Summary and conclusions	975

1 Introduction

The performance of tokamaks and other toroidal magnetic fusion devices depends crucially on the dynamics of the boundary region, i.e., the transition region from the hot core plasma through the separatrix to the material surface of the first wall, as shown in Fig. 1. Plasma turbulence, and the resulting anomalous cross-field plasma transport, are physical processes in the boundary region, affecting both core plasma confinement [e.g. high confinement mode (H-mode) and Edge Localized Modes (ELMs)], the density limit, and plasma-wall interactions [1]. The plasma boundary region has a number of physics attributes which make it quite distinct from the core: relatively low temperature, large radial gradients, and high neutral-gas and impurity densities, *proximity of open and closed flux surfaces, presence of X-point and sheath physics in the Scrape-Off-Layer (SOL)*. The large radial gradients tend to drive turbulent fluctuations which are a larger percentage of background values than in the core plasma.

Strong boundary turbulence has been observed in nearly all magnetic confinement devices [2, 3]. There exist many experimental turbulence measurements in the pedestal region and in the SOL. Common diagnostics include electrostatic probes, reflectometry, phase contrast imaging (PCI), Beam Emission Spectroscopy (BES), and Gas Puff Imaging (GPI) [2]. Observed boundary turbulence has many common features, and a great deal of experimental data has been obtained over the past 20 years on e.g. fluctuation levels, spectra, correlation lengths, and scalings, but until recently this data could not be understood from first principles. The reason is simple. The diagnostics typically are limited either to local measurements in space or to particular turbulence quantities with certain working assumptions [2]. Predictive simulation of boundary turbulence from fundamental physics models is therefore an important but daunting challenge owing to the special properties of the boundary plasma, its importance to an overall understanding of fusion plasmas, and the vast range of relevant spatial and temporal scales. A critical task is to demonstrate that simulations are able to reproduce the phenomena observed in real magnetic confinement devices. With the recent development of three dimensional (3D) non-linear codes, such as BOUT, it has become possible to make a direct computation of boundary turbulence, and validating these codes with experiments has since begun [4–11]. Using well benchmarked codes at the location of a particular measurement, boundary turbulence simulations are able to validate diagnostic tools and

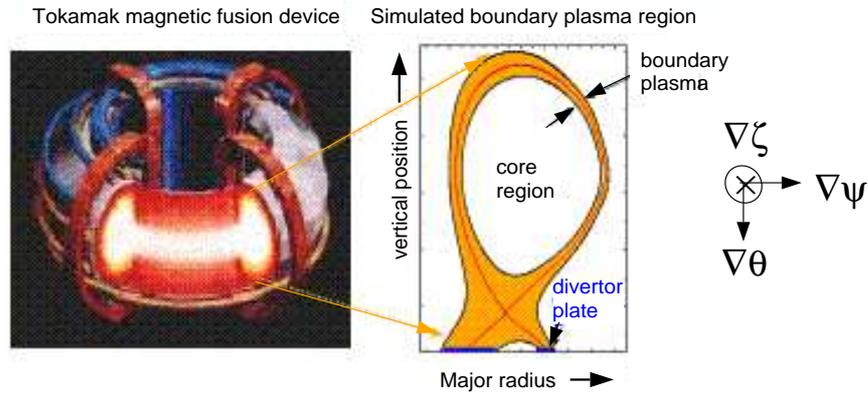


Figure 1: (color online). Schematic views of tokamak and boundary plasma region (magnetic separatrix is red line) from [38].

patch the experimental measurements together, yielding global understanding of boundary turbulence dynamics, and, most importantly, leading to scientific discoveries.

The BOUT code is a nonlinear initial-value two-fluid electromagnetic turbulence code in boundary plasmas which spans the separatrix [4, 12]. This 3-dimensional (ψ, θ, ζ) code represents configuration space via a grid in poloidal magnetic flux (ψ), poloidal angle (θ) and toroidal angle (ζ). The geometry can be a circular annulus or that of a diverted tokamak and includes boundary conditions for both closed magnetic flux surfaces and open field lines. The same set of fluid moment equations for plasma vorticity, density, ion and electron temperature and parallel momentum is discretized for both geometries. The equations are solved via a Method-of-Lines approach and an implicit backward-differencing scheme using a Newton-Krylov iteration to advance the system in time via a fully implicit Newton-Krylov solver PVODE [13]. The spatial derivatives are discretized with finite differences. A fourth-order upwinding algorithm is used for nonlinear convection, and a second-order central difference scheme is used for the rest. Boundary conditions at conducting material surfaces are implemented on the plasma side of the sheath. The 3D BOUT code is parallelized based on a domain decomposition model by implementing message passing between multiple processors by using the MPI package [14]. In order to investigate boundary turbulence, BOUT is able to couple to the edge plasma transport code UEDGE [15], and MHD equilibrium code EFIT [16] and Corsica [17] to get the realistic X-point divertor magnetic geometry and plasma profiles.

BOUT contains much of the relevant physics for the edge barrier problem for the experimentally relevant X-point divertor geometry. The calculations were carried out to validate experimental measurements and simultaneously to provide consistent understanding of boundary turbulent dynamics. Encouraging results have been obtained using measured plasma profiles in current generations of major fusion devices such as DIII-D, C-Mod and NSTX. The resistive X-point mode has been identified in X-point divertor geometry [4, 18]. Comparison of the shifted-circle vs. X-point geometry shows the different

dominant modes and turbulence fluctuation levels [4]. The poloidal fluctuation phase velocity shows experimentally observed structure across the separatrix in many fusion devices [19]. The fluctuation phase velocity is larger than $\mathbf{E} \times \mathbf{B}$ velocity. The Quasi-Coherent mode is believed to be responsible for the high energy confinement (H-mode), yet acceptably low particle (impurity) confinement in the Alcator C-Mod high density plasma regime. The experimentally measured dispersion and mode stability is in good agreement with the resistive ballooning X-point mode predicted by the BOUT code [20]. A strong poloidal asymmetry of particle flux in the proximity of the separatrix may explain the paradox of the JET probe measurement of the particle flux when comparisons of the limiter vs. divertor experiments had been made [19]. BOUT simulations performed with the measured discharge parameters show a Geodesic-Acoustic Mode (GAM) oscillation at the experimentally observed frequency [21]. Our L-H transition with simple sources added shows transitions with resistive X-point modes dominating L-mode and the levels of turbulence are similar to experimental measurements [19]. Blobs have also been clearly identified from BOUT runs and analysis of blob dynamics has shed insight on 3D X-point effects associated with increased convective velocity [3, 22, 23]. X-point effects can isolate blobs in the main SOL from divertor legs [24]. Simulations of C-MOD find blob-like structures with amplitudes and spatial correlation lengths comparable to those observed experimentally. BOUT simulations also provide evidence of instability and fluctuations in divertor legs that is uncorrelated with activity in the main SOL [25]. Simulations of Edge Localized Modes (ELMs) using the BOUT code find the expected peeling-ballooning mode structure and growth rates in the linear phase, followed by rapid radially outward propagation of filaments in the nonlinear phase [26]. Simulations of ELMs in DIII-D plasmas find a mode structure similar to that directly observed with fast cameras [26–28].

The BOUT project originally started in the late 1990s to simulate boundary turbulence across the magnetic separatrix. The goal of the BOUT project is the development and deployment of a user-friendly, state-of-art, nonlinear fluid turbulence capability for the analysis of boundary turbulence in a general geometry on a routine basis. Since then, the BOUT code has been further developed with emphasis on readability of the source code, modularity in physics models, functionality and/or macros for differential operators, consistency in higher-order spatial differencing, and finally thorough verification via test problems [29]. The most recent development, BOUT++, is based on an object-oriented approach using language C++. The aim of BOUT++ is to automate the common tasks needed for simulation codes, and to separate the complicated (and error-prone) details such as differential geometry, parallel communication, and file input/output from the user-specified equations to be solved. Thus the equations being solved are made clear, and can be easily changed with only minimal knowledge of the inner workings of the code. As far as possible, this allows the user to concentrate on the physics, rather than worrying about the numerics [30].

A method for obtaining a self-consistent model of edge-plasma turbulence and long-time edge profile evolution has been explored by coupling 2D edge transport code

UEDGE and 3D turbulence code BOUT via a relaxed iterative approach where each code is run on its own characteristic time scale. During each cycle of the iterative procedure, the toroidally averaged plasma profiles are evolved to steady state. A fraction of these profiles is used to update the profiles driving fluctuations in the 3D turbulence code. Likewise, a fractional update of the turbulent fluxes is provided to the transport code from the turbulence simulation [31]. The coupled transport/turbulence simulation technique provides a strategy to achieve physics-based predictions for future device performance.

The remainder of the paper is organized as follows. Section 2 presets physical models for boundary turbulence and a set of BOUT dynamical equations. The magnetic geometry and BOUT field-aligned coordinates are described in Section 3. The spatial numerical implementations is explained in Section 4. The background for understanding an implicit backward-differencing scheme using a Newton-Krylov iteration is presented in Section 5. The brief description of BOUT software suite is given in Section 6. The sample BOUT simulations results are illustrated in Section 7. Finally, a summary of this paper is presented in Section 8.

2 BOUT dynamical equations

2.1 BOUT plasma equations

In the boundary plasma, the application of a fluid model is reasonable in part because of the low temperature and thus high collisionality along the magnetic field. Further, the dominant modes in our simulations are in the long-wavelength regime, $k_{\perp}\rho_j \ll 1$, so the perpendicular motion can also be described by a fluid approach. The small parameters to define the ordering are:

$$\beta = 8\pi(p_j + p_e)/B^2 \ll 1, \quad \delta_j = \rho_j/L_{\perp} \sim k_{\perp}\rho_j \ll 1, \quad \Delta_j = \lambda_j/L_{\parallel} \sim k_{\parallel}\lambda_j \ll 1, \quad k_{\perp} \ll k_{\parallel}.$$

Here k_{\perp} and k_{\parallel} are the components of the fluctuation wave vector perpendicular and parallel to magnetic field, respectively. As usual, p_j is pressure, $\rho_j = v_{Ti}/\omega_{cj}$ is gyroradius, and $\lambda_j = v_{Ti}/\nu_j$ is the mean-free path with $v_{Ti} = \sqrt{2T_j/M_j}$, the thermal speed, ω_{cj} the gyrofrequency, ν_j the characteristic collision frequency, T_j temperature, M_j mass for species j ($j = i, e$). The magnitude of magnetic field is denoted by B . For application to micro-turbulence in tokamak edge transport barriers, an additional ordering $\delta_v \equiv v/v_{th,i}$ is introduced: the ratio of plasma species flow velocities to ion thermal velocity. The short mean-free path description of magnetized plasma was originally formulated by Braginskii assuming a MHD ordering $\delta_v \simeq 1$ [32]. Mikhailovskii and Tsypin first adopted a drift-ordering to give an approximate set of two-fluid equations [33], and Simakov and Catto completed the derivation with corrections [34]. The drift-ordering is defined as $v \simeq v_{pi}$ with $\delta_v \ll 1$ and the ion heat flux divided by ion density will be on the same order as the diamagnetic drift velocity v_{pi} . The end result is the ‘‘parallel’’ viscous stress tensor $\overleftrightarrow{\pi}_{cj}$ being modified. Thus an appropriate set of equations to describe the turbulence

is given by a seven-field model obtained by reduction of the Braginskii equations [4] in toroidal geometry based on drift-wave ordering with sources and sinks added:

$$\begin{aligned}
 & \frac{\partial V_{\parallel e}}{\partial t} + (\mathbf{V}_E + V_{\parallel e} \mathbf{b}_0) \cdot \nabla V_{\parallel e} \\
 = & -\frac{e}{m_e} E_{\parallel} - \frac{1}{N_i m_e} (T_e \partial_{\parallel} N_i + 1.71 N_i \partial_{\parallel} T_e) + 0.51 \nu_{ei} (V_{\parallel i} - V_{\parallel e}) \\
 & - \frac{1}{N_i m_e} \frac{2}{3} B^{3/2} \partial_{\parallel} (B^{-3/2} \pi_{ce}) + \frac{S_{\parallel e}^m}{N_i m_e} - \frac{S_e^{cx} + S_e^p}{N_i} V_{\parallel e}, \tag{2.1}
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial \omega}{\partial t} + (\mathbf{V}_E + V_{\parallel i} \mathbf{b}_0) \cdot \nabla \omega \\
 = & (2\omega_{ci}) \mathbf{b}_0 \times \kappa \cdot \left(\nabla P + \frac{1}{6} \nabla \pi_{ci} \right) + N_i Z_i e \frac{4\pi V_A^2}{c^2} \nabla_{\parallel} j_{\parallel} + \mu_{ii} \nabla_{\perp}^2 \omega \\
 & - (B\omega_{ci}) \nabla \cdot \left(\frac{\mathbf{b}_0}{B} \times (\mathbf{S}_e^m + \mathbf{S}_i^m) \right) - \left(\frac{S_i^{cx}}{N_i} \right) \omega \\
 & - (\omega_{ci} B) \nabla \cdot \left(\frac{S_i^{cx}}{N_i \omega_{ci} B} \right) \cdot (N_i Z_i e \nabla \phi + \nabla P_i) - \frac{1}{2} \{ N_i Z_i e \mathbf{V}_{Pi} \cdot \nabla (\nabla_{\perp}^2 \phi) \\
 & - M_i \omega_{ci} \mathbf{b} \times \nabla N_i \cdot \nabla V_E^2 \} + \frac{1}{2} \{ \mathbf{V}_E \cdot \nabla (\nabla_{\perp}^2 P_i) - \nabla_{\perp}^2 (\mathbf{V}_E \cdot \nabla P_i) \}, \tag{2.2}
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial N_i}{\partial t} + (\mathbf{V}_E + V_{\parallel i} \mathbf{b}_0) \cdot \nabla N_i \\
 = & \left(\frac{2c}{eB} \right) \mathbf{b}_0 \times \kappa \cdot (\nabla P_e - N_i e \nabla \phi) + \nabla_{\parallel} \left(\frac{j_{\parallel}}{e} \right) - N_i \nabla_{\parallel} V_{\parallel i} + S_e^p, \tag{2.3}
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial T_i}{\partial t} + (\mathbf{V}_E + V_{\parallel i} \mathbf{b}_0) \cdot \nabla T_i \\
 = & \frac{4}{3} \left(\frac{cT_i}{N_i e B} \right) \mathbf{b}_0 \times \kappa \cdot (\nabla P_e - N_i e \nabla \phi - \frac{5}{2} N_i \nabla T_i) + \frac{2}{3N_i} \nabla_{\parallel} (\kappa_{\parallel i}^c \partial_{\parallel} T_i) \\
 & - \frac{2T_i}{3N_i} \left(N_i \nabla_{\parallel} V_{\parallel i} - \frac{1}{e} \nabla_{\parallel} j_{\parallel} \right) + \frac{2}{3} \left(\frac{20}{3} \mu_{ii} \right) \nabla_{\perp}^2 T_i - \nu_I T_i + \left(\frac{2m_e}{M_i} \right) \frac{T_e - T_i}{\tau_e} + \frac{2S_e^E}{3N_i}, \tag{2.4}
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial T_e}{\partial t} + (\mathbf{V}_E + V_{\parallel e} \mathbf{b}_0) \cdot \nabla T_e \\
 = & \frac{4}{3} \left(\frac{cT_e}{N_i e B} \right) \mathbf{b}_0 \times \kappa \cdot \left(\nabla P_e - N_i e \nabla \phi + \frac{5}{2} N_i \nabla T_e \right) + \frac{2}{3N_i} \nabla_{\parallel} (\kappa_{\parallel e}^c \partial_{\parallel} T_e) \\
 & + \frac{2\eta_{\parallel}}{3N_i} j_{\parallel}^2 - \frac{2T_e}{3} \nabla_{\parallel} V_{\parallel e} + 0.71 \frac{2T_e}{3N_i e} \nabla_{\parallel} j_{\parallel} - \nu_I \left(T_e + \frac{2}{3} W_I \right) - \left(\frac{2m_e}{M_i} \right) \frac{T_e - T_i}{\tau_e} + \frac{2S_e^E}{3N_i}, \tag{2.5}
 \end{aligned}$$

$$\begin{aligned}
 & \frac{\partial V_{\parallel i}}{\partial t} + (\mathbf{V}_E + V_{\parallel i} \mathbf{b}_0) \cdot \nabla V_{\parallel i} \\
 = & -\frac{1}{N_i M_i} \partial_{\parallel} P - \frac{2}{3} \frac{1}{N_i M_i} B^{3/2} \partial_{\parallel} (B^{-3/2} \pi_{ci}) + \frac{S_{\parallel i}^m + S_{\parallel e}^m}{N_i M_i} - \frac{S_i^{cx} + S_i^p}{N_i} V_{\parallel i}. \tag{2.6}
 \end{aligned}$$

Also, the auxiliary variables, the vorticity ϖ , the "parallel" viscous stress tensor $\pi_{ci,e}$, the parallel electric field, and the parallel Ampère's law are given by the following set of equations:

$$\nabla_{\perp}^2 A_{\parallel} = - \left(\frac{4\pi}{c} \right) j_{\parallel}, \quad (2.7)$$

$$E_{\parallel} = -\partial_{\parallel}\phi - \left(\frac{1}{c} \right) \frac{\partial A_{\parallel}}{\partial t}, \quad (2.8)$$

$$\varpi = N_i Z_i e \nabla_{\perp}^2 \phi + N_i Z_i e \nabla_{\perp} \phi \cdot \nabla_{\perp} \ln N_i + \nabla_{\perp}^2 P_i, \quad (2.9)$$

$$\pi_{ci,e} \simeq (P_{\parallel} - P_{\perp})_{i,e} = \eta_{i,e}^0 \left((\mathbf{V}_{\mathbf{E}} + \mathbf{V}_{\mathbf{P}_{i,e}}) \cdot \boldsymbol{\kappa} - (2/\sqrt{B}) \partial_{\parallel} (\sqrt{B} V_{\parallel i,e}) \right). \quad (2.10)$$

Definitions of various quantities associated with plasma physics are as follows:

$$\begin{aligned} \mathbf{V}_{\mathbf{E}} &= c \mathbf{b}_0 \times \nabla_{\perp} \phi / B, & \mathbf{V}_{\mathbf{P}_{i,e}} &= c \mathbf{b}_0 \times \nabla_{\perp} P_{i,e} / N_i Z_i e B, \\ \tilde{\mathbf{B}} &= \nabla A_{\parallel} \times \mathbf{b}_0, & \mu_{ii} &= \frac{3}{10} v_{ii} \rho_i^2, & \eta_i^0 &= 0.96 P_i \tau_i, \\ \eta_e^0 &= 0.73 P_e \tau_e, & \omega_{cj} &= \frac{Z_j e B}{M_j c}, & V_{Aj} &= \frac{B}{\sqrt{4\pi N_j M_j}}. \end{aligned}$$

Here $\nabla_{\parallel} F = B \partial_{\parallel} (F/B)$ for any F ,

$$\partial_{\parallel} = \partial_{\parallel}^0 + \tilde{\mathbf{b}} \cdot \nabla, \quad \tilde{\mathbf{b}} = \tilde{\mathbf{B}}/B, \quad \partial_{\parallel}^0 = \mathbf{b}_0 \cdot \nabla, \quad \boldsymbol{\kappa} = \mathbf{b}_0 \cdot \nabla \mathbf{b}_0.$$

The symbol tilde represents the fluctuation quantities. Also, μ_{ii} , μ_{\parallel} , and χ_{\parallel}^c are the classical diffusion coefficients, and v_{ei} is electron collision frequency. Except for parallel viscous damping, magnetic pumping terms, and source and sink terms, similar equations are derived by Zeiler *et al.* [35]. Parallel electron viscous damping is important as it smoothes the high- k_{\parallel} oscillations near the X-point. The ion temperature equation is important for proper determination of the fluctuating electric fields (because of the ion diamagnetic drift); also it may introduce the η_i -mode in the inner edge region [36, 37]. The last two group terms in curved brackets on the right-hand-side of Eq. (2.2) are the part of the lowest-order nonlinear convection terms after the gyroviscous cancellation in an expansion in inverse aspect ratio $\epsilon = a/R$.

Definitions of various quantities associated with neutrals are as follows:

$$\begin{aligned} v_I &= N_n \langle \sigma v \rangle_I, & v_{cx} &= N_n \langle \sigma v \rangle_{cx}, \\ S^p &= N_i v_I, & S^{cx} &= N_i v_{cx}, & W_I &\simeq 20 eV. \end{aligned}$$

Here v_I is the ionization rate, v_{cx} the charge exchange rate, W_I the average energy loss per ionization. The particle source term S^p arises from ionization of neutral gas and recombination and momentum source term S^{cx} arises from charge exchange. The external source terms are S^m for momentum, S^E for energy. The derivation neglects ion momentum source/sink due to neutrals.

2.2 Modified “parallel” viscous stress tensor $\overleftrightarrow{\pi}_{cj}$

As we show in [38], the “parallel” viscous stress tensor $\overleftrightarrow{\pi}_{cj}$ yields magnetic pumping term, which is important because it damps the plasma flow shear. The turbulence fluctuation levels and transport are in turn regulated by the shear flow, via the time-varying $E \times B$ flow shear de-correlation. However, the magnetic pumping term makes a negligible contribution to linear instability because it is on the order of Δ_j smaller than other dominant linear curvature drives, such as the first ∇P term on the right-hand-side of Eq. (2.2).

Motivated by the important observation of flow shear damping, the self-consistent expression for the ion “parallel” viscous stress tensor, $\overleftrightarrow{\pi}_{cj}$, is then carefully examined and is re-derived by Simakov and Catto in the drift-ordering as follows: [34]

$$\begin{aligned} \pi_{cj} = & \eta_j^0 \left[(\mathbf{V}_E + \mathbf{V}_{Pj}) \cdot \boldsymbol{\kappa} - \frac{2}{\sqrt{B}} \partial_{\parallel} (\sqrt{B} V_{\parallel j}) \right] \\ & + \eta_j^0 \left[\left(1.61c \frac{\mathbf{b} \times \nabla T_j}{Z_j e B} \right) \cdot \boldsymbol{\kappa} - \left(\frac{7.09}{5 p_j \sqrt{B}} \right) \partial_{\parallel} (\sqrt{B} q_{\parallel j}) \right] \\ & + \eta_j^0 \left[\left(\frac{2.44 q_{\parallel j}}{5 p_j} (2.27 \nabla_{\parallel} \ln T_j - \nabla_{\parallel} \ln p_j) \right) \right] \\ & - \eta_j^0 \left[\left(\frac{4\pi c}{B^3} \right) \mathbf{b} \times \nabla (p_i + p_e) \cdot \left(\nabla \phi + \frac{\nabla p_j}{Z_j e n} + 1.61 \frac{\nabla T_j}{Z_j e} \right) \right], \quad (2.11) \\ q_{\parallel i} = & -\kappa_{\parallel i} \nabla_{\parallel} T_i, \quad \kappa_{\parallel i} = \frac{125 p_i}{32 M_i \nu_i}. \end{aligned}$$

In particular, this expression for $\overleftrightarrow{\pi}_{cj}$ allows the neoclassical Pfirsch-Schlüter expression for $V_{\parallel i}^{PS}$ to be recovered. Comparison with Eq. (2.10) for a MHD ordering shows the difference in last three-group terms with square brackets.

2.3 Analytic neutral model

Fueling at the edge of a tokamak is a complex process and requires the use of fluid or Monte Carlo codes for the best available modeling. However, for purposes of examining the role of neutrals in pedestal density formation, it is reasonable to have an analytic neutral model that includes sufficient physics to deal with the problem. One such model is a simple fluid neutral diffusion model where ion charge exchange (CX) gives rise to diffusion and ionization to the loss of neutrals. Since neutrals do not follow field lines, a simple neutral diffusion model can be setup from neutral continuity equation, with a radial coordinate r (for length in this case) to the wall,

$$\frac{\partial}{\partial r} \lambda_{CX} v_{th,n} \frac{\partial N_n}{\partial r} = N_n \nu_I. \quad (2.12)$$

where CX gives rise to diffusion as $\Gamma_n \simeq -(T_n/M_n\nu_{cx})\nabla N_n$ and ionization to the loss of neutrals. Here $\nu_I = N_e\langle\sigma_{ion}v_{th,e}\rangle$. This gives the well known results for the spatial distribution of neutrals:

$$N_n = N_w f(\theta) \exp\left(-\frac{r-r_w}{(\lambda_I\lambda_{cx})^{1/2}}\right). \quad (2.13)$$

where N_w is the neutral density at the main chamber wall and r_w is the position of the wall. The electron ionization and ion charge exchange lengths are, respectively, $\lambda_I = v_{th,n}/\nu_I$, $\lambda_{cx} = v_{th,n}/\nu_{cx}$, $v_{th,n} = \sqrt{T_n/M_n}$. Because the charge exchange collision frequency is often the largest, the gas and ion temperature are typically assumed equal, i.e., $T_n \simeq T_i$. The model provides analytic expressions for the edge N_n profile in slab geometry with the assumption that the fueling is entirely from the plasma edge. A poloidally nonuniform source of neutrals is specified by $f(\theta)$. This simple model also allows the neutral density to adjust itself to plasma profile evolution via the electron ionization length λ_I and ion charge exchange length λ_{cx} . For a typical DIII-D L-mode plasma, the neutral density varies radially less than 30% from the wall to a few centimeters inside the last closed flux surface.

2.4 Transformation of the electron parallel momentum equation

Due to the time derivative in $E_{||}$ of Eq. (2.8) in Eq. (2.1) one needs to introduce a new variable (canonical parallel momentum with unity mass),

$$A_{j||} = V_{||e} - (e/m_e c)A_{||}, \quad (2.14)$$

to cast the equation in the form solvable by the method of lines. The parallel derivative is taken with respect to perturbed magnetic field

$$\partial_{||} = \vec{b} \cdot \nabla = \vec{b}_0 \cdot \nabla + \frac{\vec{B}}{B} \cdot \nabla = \partial_{||}^0 + \frac{\nabla A_{||} \times \vec{b}_0}{B} \cdot \nabla. \quad (2.15)$$

After combining time derivative and convection terms, the following equation

$$\frac{\partial V_{||e}}{\partial t} + \mathbf{V}_E \cdot \nabla V_{||e} = -\frac{e}{m_e} E_{||}, \quad (2.16)$$

leads to

$$\frac{\partial}{\partial t} \left(V_{||e} - \frac{e}{m_e c} A_{||} \right) + \mathbf{V}_E \cdot \nabla \left(V_{||e} - \frac{e}{m_e c} A_{||} \right) = \frac{e}{m_e} \partial_{||}^0 \phi. \quad (2.17)$$

Denoting $A_{j||} = V_{||e} - (e/m_e c)A_{||}$ and the original equation (2.1) can be rewritten as

$$\begin{aligned} & \frac{\partial A_{j||}}{\partial t} + \vec{V}_E \cdot \nabla A_{j||} + V_{||e} \vec{b}_0 \cdot \nabla V_{||e} \\ &= \frac{e}{m_e} \partial_{||}^0 \phi - \frac{1}{N_i m_e} (T_e \partial_{||} N_i + 1.71 n_e \partial_{||} T_e) + 0.51 \nu_{ei} (V_{||i} - V_{||e}) \\ & \quad - \frac{1}{N_i m_e} \frac{2}{3} B^{3/2} \partial_{||} (B^{-3/2} (P_{||e} - P_{\perp e})) + \frac{S_{||e}^m}{N_i m_e} - \frac{S_s^{cx} + S_e^p}{N_i} V_{||e}. \end{aligned} \quad (2.18)$$

Here in Eq. (2.18) and in Eq. (2.1) the higher order nonlinear contribution due to the time variation of an unit direction vector of magnetic field has been neglected (term $(\partial \vec{b} / \partial t + \vec{V}_E \cdot \nabla \vec{b}) \cdot \vec{V}_{||e}$ in [34]).

Accordingly Eq. (2.7) becomes the Helmholtz equation for the $A_{||}$

$$\nabla_{\perp}^2 A_{||} - \frac{\omega_{pe}^2}{c^2} A_{||} = \frac{4\pi}{c} Z_i e N_i (A_{j||} - V_{||i}), \quad (2.19)$$

So solving it from given $A_{j||}$ and $V_{||i}$ one can find $A_{||}$, and then finds $V_{||e}$.

2.5 Simplification of vorticity

Vorticity is defined as

$$\omega = N_i Z_i e \nabla_{\perp}^2 \phi + Z_i e \nabla_{\perp} \phi \cdot \nabla_{\perp} N_i + \nabla_{\perp}^2 P_i. \quad (2.20)$$

Note that BOUT deals with perturbations of physical quantities, assuming that the zero order (equilibrium) terms cancel out altogether. The terms involving $\nabla_{\perp} \phi_0$ and $\nabla_{\perp} N_{i0}$ are dropped due to the large gradient scale length of equilibrium profiles. The nonlinear terms are dropped by an argument that this nonlinearity is not important for small fluctuations (e.g. $N_i/N_{i0} \ll 1$). Then what is left is

$$\omega = N_{i0} Z_i e \nabla_{\perp}^2 \phi + \nabla_{\perp}^2 P_i. \quad (2.21)$$

Again, N_{i0} can be put under ∇_{\perp} by same argument, and therefore the equation solved is

$$\nabla_{\perp}^2 \left(Z_i e \phi + \frac{P_i}{N_{i0}} \right) = \frac{\omega}{N_{i0}}. \quad (2.22)$$

3 Magnetic geometry

3.1 Convention for magnetic field and its sign

In a axisymmetric toroidal system, the magnetic field can be expressed as

$$\mathbf{B} = I(\psi) \nabla \zeta + \nabla \zeta \times \nabla \psi, \quad (3.1)$$

where ψ is the poloidal flux, θ is the poloidal angle-like coordinate, and ζ is the toroidal angle. Here, $I(\psi) = RB_t$, R is the tokamak major radius and B_t is the toroidal magnetic field. The two important geometrical parameters are: the curvature, κ , and the local pitch, $\nu(\psi, \theta) = I(\psi)\mathcal{J}/R^2$. The local pitch $\nu(\psi, \theta)$ is related to the MHD safety q by

$$q(\psi) = 2\pi^{-1} \oint \nu(\psi, \theta) d\theta$$

in the closed flux surface region, and

$$q(\psi) = 2\pi^{-1} \int_{inboard}^{outboard} \nu(\psi, \theta) d\theta$$

in the scrape-off-layer. Here $\mathcal{J} = (\nabla\psi \times \nabla\theta \cdot \nabla\zeta)^{-1}$ is the coordinate Jacobian, R is the major radius, and Z is the vertical position.

In our notation ζ is the geometric toroidal angle as shown in Fig. 1. Positive B_t is in the ζ direction, i.e., counter-clock-wise (looking from the top). For negative B_t , which is considered the “normal” case, the ion $\vec{\nabla}B$ drift is down [39]. For the poloidal component, B_p , the positive sign by convention corresponds to the direction from the inner plate to the outer one.

3.2 The usual flux coordinates

For such an axisymmetric equilibrium the metric coefficients are only functions of ψ and θ . Two spatial differential operators appear in the equations given as following: ∇_{\parallel} and ∇_{\perp}^2 , namely,

$$\nabla_{\parallel} = \mathbf{b}_0 \cdot \nabla = \frac{1}{\mathcal{J}B} \frac{\partial}{\partial\theta} + \frac{I}{BR^2} \frac{\partial}{\partial\zeta} = \frac{B_p}{hB} \frac{\partial}{\partial\theta} + \frac{B_t}{RB} \frac{\partial}{\partial\zeta}, \quad (3.2)$$

$$\nabla_{\perp}^2 \Phi = -\nabla \cdot [\mathbf{b} \times (\mathbf{b} \times \nabla \Phi)] = \nabla^2 \Phi - (\nabla \cdot \mathbf{b})(\mathbf{b} \cdot \nabla \Phi) - \nabla_{\parallel}^2. \quad (3.3)$$

If we use the usual flux coordinates (ψ, θ, ζ) and study the mode with

$$\rho_j \nabla_{\parallel} \simeq 0, \quad \rho_j \nabla_{\perp} \simeq 1,$$

since

$$\left| \rho_j \frac{B_p}{hB} \frac{\partial}{\partial\theta} \right| \simeq \left| \rho_j \frac{B_t}{RB} \frac{\partial}{\partial\zeta} \right| \simeq \varepsilon/q, \quad \text{for } k_{\theta} \rho_j \simeq 1.$$

$\nabla_{\parallel} \simeq 0$ depends on the cancelation of two finite and almost equal numbers. Consequently, it is difficult to obtain accurate numerical solutions.

3.3 Field-aligned coordinates

In order to efficiently simulate turbulence with short perpendicular wavenumbers $k_{\parallel} \ll k_{\perp}$, we choose field-aligned coordinates [4, 40–42], (x, y, z) , which are related to the usual flux coordinates (ψ, θ, ζ) by the relations

$$\begin{aligned} x &= \psi - \psi_s, \\ y &= \theta, \\ z &= \zeta - \int_{\theta_0}^{\theta} v(x, y) dy. \end{aligned} \quad (3.4)$$

In the field-aligned coordinates, the parallel differential operator is simple, involving only one coordinate y

$$\partial_{\parallel}^0 = \mathbf{b}_0 \cdot \nabla_{\parallel} = \left(\frac{B_p}{hB} \right) \frac{\partial}{\partial y} = \mathcal{J}_{\parallel} \frac{\partial}{\partial y}. \quad (3.5)$$

which requires a few grid points. However, magnetic shear leads to strong deformation of coordinate cells in the plane perpendicular to the field, and spatial discretization of radial derivative has to be taken care with special methods [4, 40, 41]. To remedy this we introduce a radial difference procedure in dual sets of coordinate systems. Even though the simulation data resides on the field-aligned coordinates, the radial difference will be computed in the usual flux (ψ, θ, ζ) coordinates. Thus a high order interpolation scheme is needed to map data back and forth between the field-aligned coordinates and the usual flux (ψ, θ, ζ) coordinates.

The derivatives are obtained from the chain rule as follows:

$$\frac{\partial}{\partial \psi} = \frac{\partial}{\partial x} - I \frac{\partial}{\partial z}, \quad (3.6)$$

$$\frac{\partial}{\partial \theta} = \frac{\partial}{\partial y} - v(x, y) \frac{\partial}{\partial z}, \quad (3.7)$$

$$\frac{\partial}{\partial \zeta} = \frac{\partial}{\partial z}, \quad (3.8)$$

$$I = \left(\int_{y_0}^y \frac{\partial v(x, y)}{\partial \psi} dy \right). \quad (3.9)$$

The coordinate Jacobian and metric coefficients are defined as following:

$$\mathcal{J} = (\nabla \psi \times \nabla \theta \cdot \nabla \zeta)^{-1} = \frac{h}{B_p}, \quad (3.10)$$

$$h = \sqrt{Z_{\theta}^2 + R_{\theta}^2}, \quad (3.11)$$

$$\mathcal{J}_{11} = |\nabla x|^2 = \frac{R^2}{\mathcal{J}^2} (Z_{\theta}^2 + R_{\theta}^2), \quad (3.12)$$

$$\mathcal{J}_{12} = \mathcal{J}_{21} = \nabla x \cdot \nabla y = -\frac{R^2}{\mathcal{J}^2} (Z_{\theta} Z_{\psi} + R_{\psi} R_{\theta}), \quad (3.13)$$

$$\mathcal{J}_{22} = |\nabla y|^2 = \frac{R^2}{\mathcal{J}^2} (Z_\psi^2 + R_\psi^2), \quad (3.14)$$

$$\mathcal{J}_{33} = \frac{1}{R^2}, \quad (3.15)$$

$$\mathcal{J}_{\parallel} = \frac{B_p}{hB}. \quad (3.16)$$

Here h is the local minor radius, and y_0 is an arbitrary integration parameter. The disadvantage of this choice of coordinates is that the Jacobian diverges near the X-point as $B_p \rightarrow 0$. Therefore a better set of coordinates is needed for X-point divertor geometry.

3.4 Twist-shift boundary conditions

In the field-aligned coordinates the y coordinate is no longer periodic, but pseudo-periodic, because it is the coordinate along the field line and it carries with it a toroidal displacement in the binormal direction z . Therefore the twist-shift boundary conditions are applied in the poloidal direction inside the separatrix and periodic boundary conditions are applied in the binormal/toroidal direction:

$$f(\psi, \theta + 2\pi, \zeta) = f(\psi, \theta, \zeta) \longrightarrow f\left(x, y + 2\pi, z - \oint v dy\right) = f(x, y, z), \quad (3.17)$$

$$f(\psi, \theta, \zeta + 2\pi) = f(\psi, \theta, \zeta) \longrightarrow f(x, y, z + 2\pi) = f(x, y, z), \quad (3.18)$$

where the integral $\oint v(x, y) dy$ represents a global shift by the periodicity constraint in the field-aligning transformation. If the simulation domain is the entire flux surface (the parallelogram ABEF in Fig. 2, $0 \leq y \leq 2\pi$, $0 \leq z \leq 2\pi$), the complete set of the Fourier modes n and m are kept.

3.5 Annular toroidal wedge

For efficient simulations of turbulence with the high toroidal mode number n ($n \gg 1$), in best practice a truncated computational domain on the full torus down to a toroidal wedge (the parallelogram AB_1E_1F) is generally used, as shown in Fig. 2. To ensure the toroidal periodicity, the full torus has to be divided into an integer Δn equal parts (toroidal wedges) and enforce periodicity on each of them. However, after one poloidal cycle along the field line, the end of the field line is shifted by a global pitch $\Delta\zeta \equiv \oint v(\psi, \theta) d\theta$, in general it may fall into a different but an identical toroidal wedge, so the poloidal periodicity must be enforced after each such cycle. The periodic boundary conditions in a annular toroidal wedge then become:

$$f(\psi, \theta, \zeta + 2\pi) = f(\psi, \theta, \zeta) \longrightarrow f(x, y, z + 2\pi / \Delta n) = f(x, y, z), \quad (3.19)$$

$$f(\psi, \theta + 2\pi, \zeta) = f(\psi, \theta, \zeta) \longrightarrow f\left(x, y + 2\pi, z - \oint v dy\right) = f(x, y, z), \quad (3.20)$$

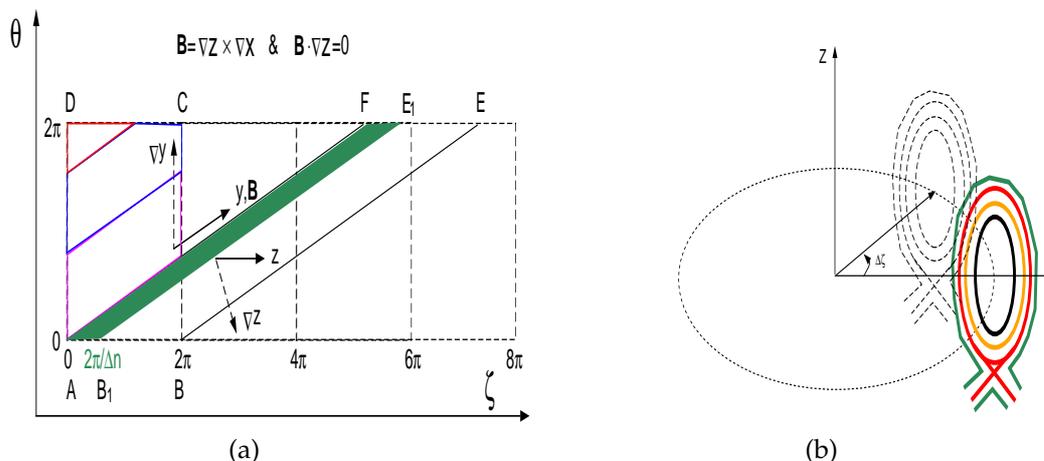


Figure 2: (color online). (a) A sketch of the field-aligned coordinates mapping from (θ, ζ) to (y, z) . The area covered by the square ABCD is for the usual flux coordinates (ψ, θ, ζ) . The area covered by parallelogram ABEF is for the field-aligned coordinates (x, y, z) . The green area covered by the parallelogram AB_1E_1F is a truncated simulation domain by the name of an annular toroidal wedge; (b) A sketch of a annular toroidal wedge. The width of the wedge $\Delta\zeta = 2\pi/\Delta n$, where $\Delta n \geq 1$ is an integer.

where Δn is a quantization constant, an integer. The consequence of the truncation of the computational domain is that the complete set of the n -spectrum is reduced from $n = (0, \pm 1, \pm 2, \dots)$ to $n = (0, \pm 1\Delta n, \pm 2\Delta n, \dots)$ in simulations of a annular toroidal wedge.

4 Spatial numerical implementations

4.1 Radial derivatives in field-aligned coordinates

According to the chain rule, the transformation of the radial derivative in field-aligned coordinates is given in Eq. (3.6). Due to the magnetic shear, the coordinate cell deformation occurs: a rectangle cell $(\Delta\psi \sim \Delta\zeta)$ in the flux coordinate becomes elongated in field-aligned coordinates $\Delta x \gg \Delta z$ by a factor of I due to secular poloidal displacement of the two field lines, as shown in Fig. 3. The best way to handle this is to use finite Fourier transforms which is exact up to machine accuracy. Applying Fourier transform in z leads to

$$\frac{\partial}{\partial\psi} f_{kz}(x, y) = \frac{\partial}{\partial x} f_{kz}(x, y) - (ik_z I) f_{kz}(x, y), \quad (4.1)$$

where I is defined in Eq. (3.9) and can be pre-computed from magnetic geometry. However a preferred method is to (1) shift the Fourier transformed variable by a phase factor $\exp[-ik_z \int_{y_0}^y v(\psi, y) dy]$, and then (2) perform a finite difference on the transformed variable $f_{kz}(x, y) \exp[-ik_z \int_{y_0}^y v(\psi, y) dy]$ in the usual flux coordinates where there is no cell deformation. Even though theoretically they are identical when $\Delta\psi \rightarrow 0$, this method has

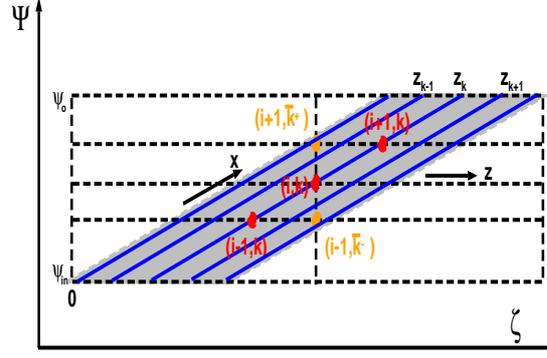


Figure 3: (color online). A sketch of adjacent grid-points mapping from the field-aligned coordinates $(x_{i-1}, y_j, z_k) \rightarrow (x_i, y_j, z_k) \rightarrow (x_{i+1}, y_j, z_k)$ to the usual flux surface coordinates $(\psi_{i-1}, \theta_j, \zeta) \rightarrow (\psi_i, \theta_j, \zeta) \rightarrow (\psi_{i+1}, \theta_j, \zeta)$.

a advantage for a consistency in numerical approximations to match the twist-shifted boundary condition at the branch cut due to finite radial grid spacing $\Delta\psi$ as following:

$$\begin{aligned} \mathcal{I}_{kz} &\equiv \frac{\partial}{\partial\psi} \exp \left[-ik_z \int_{y_0}^y v(\psi, y) dy \right] \\ &\simeq \frac{\exp \left[-ik_z \int_{y_0}^y v(\psi + \Delta\psi, y) dy \right] - \exp \left[-ik_z \int_{y_0}^y v(\psi, y) dy \right]}{\Delta\psi} \\ &\xrightarrow{\Delta\psi \rightarrow 0} -ik_z I. \end{aligned} \quad (4.2)$$

In other words, the variable $-ik_z I$ in Eq. (4.1) should be numerically pre-computed as one complex variable \mathcal{I}_{kz} defined as in Eq. (4.2), instead of as $(-ik_z)$ times I .

4.2 Inversion of the elliptic operators

The field equations (2.19) for A_{\parallel} and (2.22) for ϕ are elliptic partial differential equations of second order in divergence form $\nabla^2 u - \alpha u = f$. When α is equal to zero a ‘‘pure’’ Laplacian or Poisson equation results and when α is greater than zero a so called Helmholtz equation is produced. The modern numerical schemes to efficiently solve this type of equations on massive parallel computer platforms are iterative techniques based on the multigrid methods [43–45]. However, a direct solver is still preferred for its simplicity whenever possible.

4.2.1 Inversion of potential vorticity

The vorticity equation is defined in Eq. (2.22). Using $\mathcal{F} \equiv Z_i e\phi + P_i / N_{i0}$ and the differential operator given in the Appendix (A.4), and applying Fourier transform in z leads to

$$(RB_{\theta})^2 \left\{ \frac{\partial^2 \mathcal{F}_{kz}}{\partial x^2} + 2\mathcal{I}_{kz} \frac{\partial \mathcal{F}_{kz}}{\partial x} + \left[\mathcal{I}_{kz}^2 - k_z^2 \frac{B^2}{(RB_{\theta})^4} \right] \mathcal{F}_{kz} \right\} = \frac{\omega_k}{\hat{N}_{i0}}. \quad (4.3)$$

Here \mathcal{I}_{kz} is defined in Eq. (4.2). Solving the ordinary differential equation (ODE) using a tridiagonal linear solver yields $\mathcal{F}_{kz}(x, y)$, then inverse Fourier transform yields $\mathcal{F}(x, y, z)$ in the field-aligned coordinates, from the latter $Z_i e\phi = \mathcal{F}(x, y, z) - P_i / N_{i0}$. Note that in this procedure the radial boundary conditions for N_i , T_i , and ϕ become linked together.

4.2.2 Inversion of A_{\parallel}

Similarly, the Helmholtz equation for A_{\parallel} from Eq. (2.19) in Fourier space can be written as

$$(RB_{\theta}) \left\{ \frac{\partial^2 A_{\parallel kz}}{\partial x^2} + 2\mathcal{I}_{kz} \frac{\partial A_{\parallel kz}}{\partial x} + \left[\mathcal{I}_{kz}^2 - k_z^2 \frac{B^2}{(RB_{\theta})^4} \right] A_{\parallel kz} - \frac{\omega_{pe}^2}{c^2} A_{\parallel kz} \right\} = \frac{4\pi}{c} N_{i0} Z_i e \left(A_{j\parallel} - V_{\parallel i} \right)_{kz}. \quad (4.4)$$

The solution procedure is same as that for the vorticity.

4.3 Numerical implementation of boundary conditions

4.3.1 Toroidal (z) boundary conditions

To ensure the toroidal periodicity, a full torus has to be divided into an integer Δn equal parts (toroidal wedges) and enforce periodicity on each of them.

$$F(x, y_k, z_{k+N}) = F(x, y_k, z_k), \quad (4.5)$$

$$z_{k+N} = z_k + L_z = z_k + N_z \Delta z, \quad (4.6)$$

where L_z is the toroidal length of the wedges, N_z is the number of nodal points, and Δz is the cell width.

4.3.2 Parallel (y) boundary conditions

The boundary conditions for Φ is the sheath boundary conditions in y in the SOL and the private flux regions at the divertor plates, pseudo-periodic in y in "Edge" (the outer part of the closed flux region inside of separatrix), as shown in Fig. 4.

• **Twist-shift Edge boundary conditions.** After one poloidal cycle along the field line, the end of the field line is shifted by a global pitch $\oint v(\psi, \theta) dy$ in the binormal direction z , in general it may fall into a different but an identical toroidal wedge, so the poloidal periodicity must be enforced after each such cycle. A sketch of such procedures is pictorially shown in Fig. 5(a). The pseudo-periodic boundary conditions in an annular toroidal wedge then become:

$$F(x, y_{k+N}, \bar{z}_{k+N}) = F(x, y_k, z_k), \quad (4.7)$$

$$\bar{z}_{k+N} = \begin{cases} z_k - \left[\oint v(x, y) dy \right] \% L_z, & z_k > \left[\oint v(x, y) dy \right] \% L_z, \\ z_k - \left[\oint v(x, y) dy \right] \% L_z + L_z, & z_k < \left[\oint v(x, y) dy \right] \% L_z, \end{cases} \quad (4.8)$$

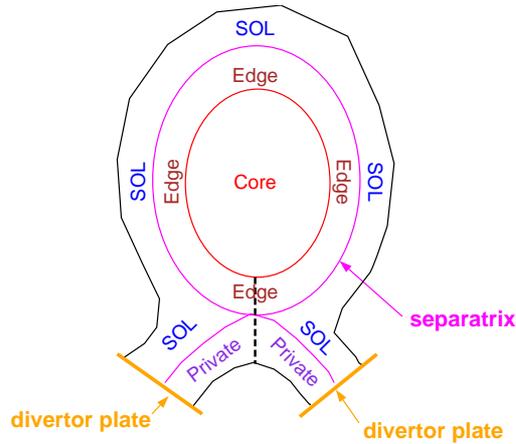


Figure 4: (color online). The radial-poloidal plane is divided into three main regions: outer part of core (Edge), SOL, private flux region and divertor plates.

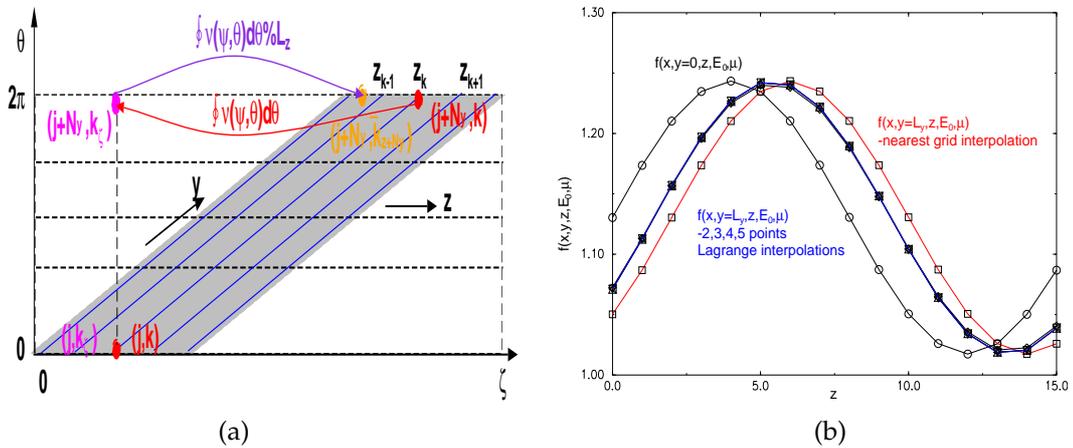


Figure 5: (color online). (a) A sketch of poloidal periodic grid-points mapping from the field-aligned coordinates $(x_i, y_j + 2\pi, \bar{z}_k) \rightarrow (x_i, y_j, z_k)$ to the usual flux surface coordinates $(\psi, \theta_j + 2\pi, \zeta) \rightarrow (\psi, \theta_j, \zeta_k)$, where $\bar{z}_k = z_k - \oint v(\psi, \theta) d\theta$; (b) test results of twist-shifted periodic y-boundary using Lagrange interpolation scheme.

where the integral $\oint v(x, y) dy$ represents a global shift by the periodicity constraint in the field-aligning transformation. The symbol % represents the modulus operator, and the expression $x \% y$ produces the remainder when x is divided by y , and thus is zero when y divides x exactly.

The consequence of these shifts is that there must be interpolation in z_k to the fact that the field-aligning transformation itself involves a shift and in general \hat{z}_k after the shift does not fall exactly on a nodal point in z_k . Therefore testing must be done to show that to which order of interpolation it makes no difference in the turbulence. Of course, finite Fourier transforms are exact up to machine accuracy. Using Lagrange interpolation scheme [46], we found that for a longest wavelength in z -direction, the 2-, 3-, 4-, and 5-points interpolation yield almost the same result, as shown in Fig. 5(b).

• **Sheath boundary conditions.** An electrostatic sheath will form at any plasma boundary and acts to filter all but the high energy electrons while attracting ions, controlling the particle and energy flux leaving the plasma. Each ion-electron pair crossing the sheath convects to the surface a quantity of energy which is conventionally described using a total sheath heat transmission coefficient where [39] $q_{se} = \gamma_e k T_e \Gamma_{se}$ with q_{se} total heat flux at the sheath edge, T_e the temperature, $\Gamma_{se} \equiv N_i c_{se}$ the sheath edge particle flux and where

$$\gamma_e = 2.5 \frac{T_i}{T_e} + \frac{2}{1 - \delta_e} - 0.5 \ln \left[\left(2\pi \frac{m_e}{M_i} \right) \left(1 + \frac{T_i}{T_e} \right) \frac{2}{(1 - \delta_e)^2} \right] \quad (4.9)$$

with δ_e the secondary electron emission coefficient and T_i the ion temperature. Therefore in the SOL and private flux region, the divertor plate boundary conditions are

$$V_j = c_{se} = \sqrt{\frac{T_i + T_e}{M_j}}, \quad (4.10)$$

$$j_{\parallel}^{el} = N_i e \left[c_{se} - \frac{v_{Te}}{2\sqrt{\pi}} \exp\left(-\frac{e\phi}{T_e}\right) \right], \quad (4.11)$$

$$q_{se} = -\kappa_{\parallel e} \partial_{\parallel} T_e = \gamma_e N_i T_e c_{se}, \quad (4.12)$$

$$q_{si} = -\kappa_{\parallel i} \partial_{\parallel} T_i = \gamma_i N_i T_i c_{se}, \quad (4.13)$$

$$\partial_{\parallel} \omega = 0, \quad (4.14)$$

$$\partial_{\parallel} N_i = 0, \quad (4.15)$$

where $\gamma_i \simeq 2.5$ and $\gamma_e \simeq 7$ are sheath energy transmission factors. Here we assume that a magnetic field is in the normal direction to the divertor plates. There is no boundary condition for density N_i and vorticity ω . If a boundary condition has to be imposed for numerical reasons, the zero parallel gradient is used at the sheath entrance.

5 Solving BOUT equations with PVODE

The fluid equations in Section 2 solved by BOUT can be cast in the most general form in terms of a system of time-dependent ordinary differential equations (ODEs)

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, \nabla \mathbf{u}, \nabla^2 \mathbf{u}, \dots) \quad (5.1)$$

where \mathbf{u} is the vector of unknowns at a given mesh point, and \mathbf{f} is typically called right-hand-side function (rhs-f), which involves variables \mathbf{u} at a mesh point, and spatial derivatives of variables \mathbf{u} which are computed using finite difference. The rhs-f consequently depends upon the mesh point, its close neighbors, and difference schemes used.

BOUT code presently uses the Newton-Krylov approach. This scheme is exemplified by the BDF method (for Backward Differentiation Formula). For the BDF method, the

advancement of \mathbf{u} from time level $n-1$ to n takes the form

$$\mathbf{u}_n = (\beta_1 \mathbf{u}_{n-1} + \dots + \beta_k \mathbf{u}_{n-k}) + \Delta t \gamma_0 \mathbf{f}_n. \quad (5.2)$$

The BDF method is usually solved by a Newton iteration which expands \mathbf{f}_n at iteration j as

$$\mathbf{f}(\mathbf{u}^j) \approx \mathbf{f}(\mathbf{u}^{j-1}) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} (\mathbf{u}^j - \mathbf{u}^{j-1}). \quad (5.3)$$

Eq. (5.2) then is a linear equation for \mathbf{u}_n^j which can be written as

$$(\mathbf{I}/\Delta t \gamma_0 - \mathbf{J}) \mathbf{u}_n^j = \mathbf{g} \quad (5.4)$$

where \mathbf{I} is the identity matrix, $\mathbf{J} \equiv \partial \mathbf{f} / \partial \mathbf{u}$ is the Jacobian evaluated with \mathbf{u} from a previous iteration or time step. Also, \mathbf{g} is a vector which depends on values of \mathbf{u} from the past iteration, \mathbf{u}^{j-1} , and at previous time steps as obtained from Eqs. (5.2)-(5.3). Eq. (5.4) is usually solved by an iterative method to an accuracy somewhat better than the estimated error in \mathbf{u}_{n-1} from the time advancement; this is known as an inexact Newton method. We shall use a Krylov projection method to solve the linear system provided by a fully implicit solver: PVODE [47, 48]. Although more work is required for such Newton methods per iteration, they often have superior overall performance for stiff ODEs since larger time steps can be used. We compared two methods of advancing the equations in time: one is the Adams functional iteration (equivalent to predict-corrected method for a one-step functional iteration) and the second is the inexact Newton method utilizing matrix-free Krylov projections as described. We found that the Newton-Krylov method is able to expand its time step by a factor of 70 in linear stage compared to the functional-iteration Adams method for the same accuracy [49]. However, it is only about six times more efficient due to the extra work required to expand the large time step. In the nonlinear stage of the simulation where different wave modes are strongly coupled, the Newton method reduces its time step by about 1/2 to satisfy the accuracy constraint. In fact, this simulation includes the shear in the magnetic equilibrium near the X-point which was a problem that we could not integrate successfully with the previous predictor-corrector method (a one-step functional iteration). Thus, using the Newton-Krylov method has become an essential part of our BOUT simulations.

Newton schemes that utilize a matrix-free Krylov projection method often require preconditioning [47, 48]. The procedure requires the ability to solve related linear systems $\mathbf{P}\mathbf{v} = \mathbf{h}$ with a matrix \mathbf{P} which approximates the original matrix, but is simpler to solve. By assumption, $\mathbf{P} \sim (\mathbf{I}/\Delta t \gamma_0 - \mathbf{J})$. Noting that $\mathbf{P}^{-1}\mathbf{P} = \mathbf{I}$, we may insert this product into Eq. (5.4) to form the preconditioned system

$$[(\mathbf{I}/\Delta t \gamma_0 - \mathbf{J})\mathbf{P}^{-1}](\mathbf{P}\mathbf{u}_n^j) = \mathbf{g}. \quad (5.5)$$

The new variables are $\mathbf{P}\mathbf{u}_n$, and this system is easier to solve by iterative methods such as the Krylov method since

$$[(\mathbf{I}/\Delta t \gamma_0 - \mathbf{J})\mathbf{P}^{-1}] \equiv \mathbf{A} \sim \mathbf{I}$$

is more diagonally dominant. The Krylov method does require matrix-vector products of $\mathbf{A}(\mathbf{P}\mathbf{u}_n)$, and these are done in a matrix-free manner with a finite-difference quotient approximation $\mathbf{J}\mathbf{v}$ [47]. However, we find that the implicit BOUT works well without a preconditioner, which may be related to the smaller time step required to resolve the turbulent fluctuations.

6 BOUT software design

To simulate boundary plasma turbulence and validate with the corresponding experiments, the BOUT code uses realistic X-point magnetic and plasma profiles. The background magnetic field structure is obtained from an MHD equilibrium code (usually EFIT [16]) for a typical shot. The plasma profiles are obtained by taking density and temperature as analytic fits (such as modified tanh) to Thomson scattering data or calculated from the edge transport code UEDGE [15]. For theoretical scaling studies with plasma current, the background magnetic field structure is obtained from another MHD equilibrium code, Corsica [17]. For typical DIII-D boundary plasma profiles in L-mode, the mid-plane values on the magnetic separatrix are: $T_e = 60\text{eV}$, $T_i = 240\text{eV}$, and $n_i = 6.5 \times 10^{18} / \text{m}^3$. From the given magnetic geometry and plasma profiles corresponding to a specific experimental device and discharge, the simulation is initialized with a set of small random fluctuations. The fastest growing modes dominate the initial phase of the calculation, in which the fluctuations grow at an approximately exponential rate. After this initial linear phase, the density and electrostatic potential fluctuations evolve to a saturated state with many modes. From the saturated steady state, turbulence statistical properties can be extracted from the BOUT simulations by using the correlation function analysis and validated with the various fluctuation measurements [19, 20, 38, 50]. The procedures of utilization of BOUT software suite is sketched in Fig. 6. Here EFIT and Corsica are MHD equilibrium codes for initial magnetic geometry setup for a whole device, UEDGE is an edge transport code for finer grid generation at the boundary region across the magnetic separatrix using spline and/or for plasma profiles. ELITE is a linear MHD stability code for peeling-ballooning modes [51] and BAL is a linear stability code for drift-wave-type instabilities [18] and lately 2DX [52] for BOUT benchmark studies, and GKV is a collection of IDL routines using the correlation function techniques for data analysis. Data from BOUT simulations are saved and later analyzed with the GKV and other BOUT data analysis IDL routines for post-processor to obtain fluctuation spectra, two-point correlation functions (including correlation times and lengths), bi-spectra, etc.

BOUT kernel code is a collection of subprograms that embodies physical or numerical functionalities, which includes, but is not limited to, the following: grid generation, data allocation, initial conditions, boundary conditions, field-solve for vorticity and vector potential, rhs-f evaluation, and parallelization, and interface between the BOUT data and PVODE data (which advances a vector of variables \mathbf{u} using the Newton-Krylov method). The complete BOUT code description can be found in BOUT manual [53].

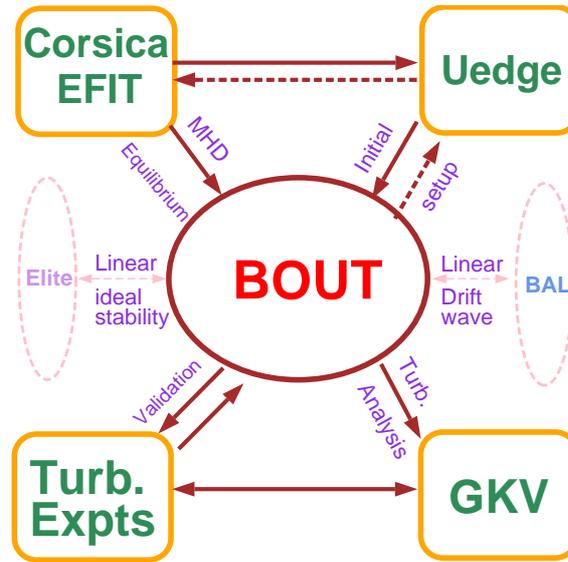


Figure 6: (color online). BOUT software suite: a suite of the codes work together to make BOUT simulation results similar to real experiments.

BOUT is parallelized via a poloidal domain decomposition model that uses the MPI (Message Passage Interface) system [14]. The parallel implementation is straightforward and efficient: one or several poloidal meshes with the entire radial-toroidal plane are stored on each processor. At the end of a time step, the data in the domain boundary planes are passed to its physically neighboring processor. Because of this parallel paradigm, the amount of message passing scales linearly with the problem size. For a typical run with 64 processors, the communication time is less than 1% [49].

7 BOUT simulation results

BOUT is an electromagnetic fluid edge turbulence code [4, 12]. The physics model is based on the Braginskii equations for a collisional plasma, and the equations are solved numerically in the real geometry of a divertor tokamak. BOUT model supports a large variety of plasma modes: ideal and resistive ballooning and kink modes, drift, shear-Alfven, sheath-driven modes and others.

With such a complex simulation software suite, as discussed in [29], two questions naturally arise: (i) whether the equations form a valid physics model for the phenomena studied, and (ii) whether the equations are solved correctly by the code. The answer to the first question is that for sufficiently collisional plasma, the collisional closure should hold and thus the model should be valid. To address the second question one needs to do thorough verification testing to make sure the numerical model can be trusted as a research tool. Four test problems have been benchmarked for verification of the BOUT

code [29]: (1) Shear Alfvén wave; (2) Resistive drift instability; (3) Resistive interchange instability; (4) Axisymmetric benchmark with UEDGE. A suite of codes are under development to extend the verification studies, and it is a step toward the creation of accepted verification standards for edge turbulence codes. These include linearized and nonlocal (e.g. separatrix-spanning) modes in axisymmetric (realistic divertor) toroidal geometry. The suite consists of (i) an initial value approach using the BOUT 3D fluid turbulence code, here run with the nonlinear terms turned off, and (ii) a new linear eigenvalue code 2DX [52] for the boundary plasma. These are so-called Verification and Validation (V&V) processes, and essentially confidence building activities [54–58].

Due to the inability to obtain an exact analytical solution, assessing the accuracy of a nonlinear turbulence code and simulations is even harder and requires: (1) a grid refinement study to show that the grid is sufficiently refined such that the solution is in the asymptotic range of convergence; (2) inter-code benchmarks; simulation results have to be verified with other turbulence codes which have the same underlying physics models and simulation parameters; (3) code verification by the method of manufactured solutions [57]. Even so, one must recognize the distinction between a numerical result which approaches an asymptotic numerical value and one which approaches the true solution. It is hoped that as the grid is refined and resolution improves that the simulated turbulence dynamics, its ensemble averages and statistics will not change much and approach an asymptotic value (i.e. the true numerical solution). There still may be error between this asymptotic value and the true physical solution to the equations. Sources of error include, but not limit to, numerical algorithms, spatial or temporal gridding, coding errors, language or compiler bugs, iterative convergence errors, computer round-off, and so forth.

In the following, two nonlinear BOUT simulation results are presented to demonstrate its capabilities, functionalities, preliminary V&V efforts (to compare trends rather than absolute values).

7.1 Density effects on tokamak edge turbulence

A series of BOUT simulations has been conducted to investigate the physical processes which limit the density in tokamak plasmas [59]. In this section, the plasma profiles are frozen, while they are evolved in Section 7.2. With poloidal flux, ψ , normalized to unity on the separatrix, we typically take the inner simulation boundary condition to be $\psi_c=0.9$ and the outer boundary at $\psi_w=1.1$. The toroidal segment is typically one tenth of the torus with full poloidal cross section. The boundary conditions for turbulence variables are homogeneous Neumann at $\psi = x_c$ and at $\psi = x_w$, sheath boundary conditions in y in the SOL and the private flux regions at the divertor plates, twist-shifted periodic in y in the closed flux region due to the choice of field-aligned coordinates, and periodic in z . However for the electrostatic potential with the toroidal mode number $n=0$ component, the boundary conditions are homogeneous Neumann at $x = x_c$ and Dirichlet at $x = x_w$. The computational mesh has 64 poloidal and 64 toroidal mesh points, and 50

radial points. The background magnetic field structure is obtained from an MHD equilibrium code (*e.g.*, EFIT [16]) for a typical discharge. The plasma profiles of density and electron temperature T_e , are analytic fits (modified tanh) to Thomson scattering data. For scaling studies with plasma density, the plasma pressure is held constant. For scaling studies with plasma current, the background magnetic field structure is obtained from the MHD equilibrium code Corsica [17]. Since there is no unstable edge localized mode (ELM) for our base case (L-mode), there is also no ELM for the density scans. Furthermore, the current gradient driven modes are explicitly turned off in this study to focus our efforts on the density effects [38]. In these simulations, a grid convergence study shows 20% variation of the peak ion heat flux over following scan: $n_\zeta = 32$, $n_\zeta = 64$, and $n_\zeta = 256$ with $n_\psi = 50$ and $n_\theta = 64$.

Simulations of turbulence in tokamak boundary plasmas show that turbulent fluctuation levels and transport increase with collisionality. As the edge density increases and the temperature decreases, BOUT simulations show that the resistive X-point mode transitions to the resistive ballooning mode, perpendicular turbulent transport approaches and finally dominates parallel classical transport, leading to substantially reduced contact with divertor plates and the destruction of the $\mathbf{E} \times \mathbf{B}$ edge shear layer; the region of high transport then extends inside the last closed magnetic flux surface. The full consequences of the large radial transport were assessed by a set of 2D UEDGE transport simulations with increasing outboard convective radial transport to mimic the BOUT results for increasing density. These simulations show that this transport can lead to an X-point MARFE when a fixed-fraction carbon impurity radiation is included [59]. BOUT further demonstrates that the current scaling appears on a plot of discharge current versus density as an abrupt increase in radial transport once $\bar{n}_e/n_G > 1$. All of these results indicate that rapid edge cooling due to large radial transport is a key physics element of the tokamak density limit. The simulation results are qualitatively consistent with experimental observations from C-mod and DIII-D [60, 61] and analytical analysis including perpendicular heat convection based on the blob heat transport model [62].

These simulations are qualitatively consistent with previous theory and simulations given by Rogers, Drake, and Zeiler (RDZ) [5], with the exception of the safety factor q -dependence in their α_d scaling. The three sets of simulations are extrapolated to compare with RDZ theory and experiments, and to check whether a density limit boundary line is crossed, as the arrows indicate in Fig. 7. In what follows, the words “agree” or disagree” is in a qualitative sense, *i.e.*, the same trend. (1). For fixed q , current I_p and pressure P , an increase in density n_e leads to a fixed α and a decrease in α_d ,

$$\alpha_d \propto \sqrt{\lambda_{mfp}^e} \propto \frac{1}{\sqrt{n_e}}.$$

In this case, the density-limit boundary is crossed, and RDZ theory, BOUT simulations and experiments agree. (2). For fixed q , temperatures T_e , T_i and density n_e , a decrease in current I_p leads to an increase in $\alpha \approx 1/I_p^2$ and constant α_d . In this case, the density limit boundary is crossed, and RDZ theory, BOUT simulations and experiments agree. (3). For

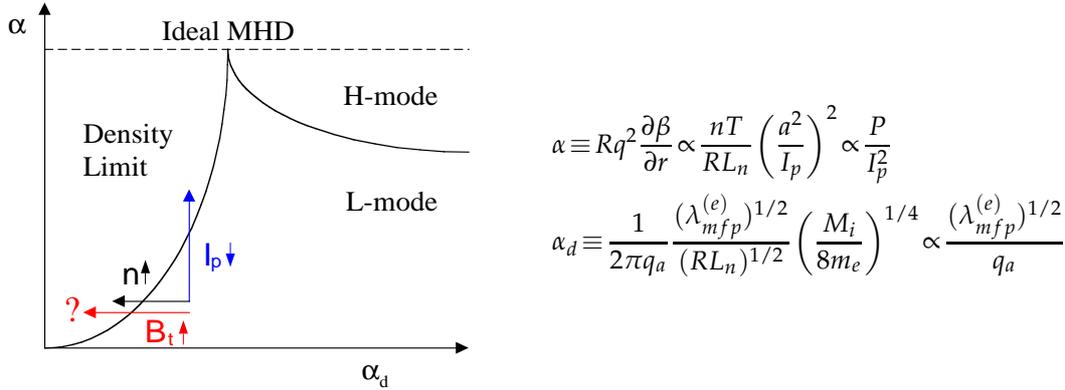


Figure 7: (color online). A sketch of edge plasma phase space from Rogers, Drake, and Zeiler theory [5].

fixed I_p , T_e , T_i and n_e , an increase in toroidal magnetic field B_t leads to a fixed α and a decrease in $\alpha_d \approx 1/q$ since $q \propto B_t$. In this case, the RDZ theory predicts a density limit, but both experiments [63] and BOUT find no transition for this case. The disagreement may be due to two important pieces of physics omitted from RDZ theory that are kept in BOUT simulations: X-point physics and SOL open-magnetic-field-line physics. X-point physics limits the mode to the outside midplane such that the parallel connection length qR is not a good measure of the parallel mode width because of $q_a \rightarrow \infty$ near the magnetic separatrix in the divertor geometry. SOL physics contributes significantly to the formation of the E_r well and our simulations show that the onset of large radial transport is associated with the destruction of the E_r well [59].

7.2 Blob dynamics and correlation analysis

For self-consistent turbulence and transport simulations with a neutral source added, as described in Section 2.3, we find that as density rises due to neutral fueling, turbulent transport increases. The same trend has been obtained with fixed plasma profiles as discussed in the previous section. The characteristics of the fluctuations also change from small scale turbulence to large density structures called blobs [3, 64]. At high density during density ramp-up simulations, we have identified convective transport by localized plasma blobs in the SOL [22, 23, 65]. Such strong intermittent edge transport has been simulated previously in a 2D slab geometry [66].

An animation (<http://www.global-sci.com/video/v4/949.mpg>) is given during density ramping, showing shear flow and blob dynamics. A simple fluid neutral diffusion model is used where ion charge exchange (CX) gives rise to diffusion and ionization to the loss of neutrals. The neutral density at the wall is $N_w = 1 \times 10^{11} \text{ cm}^{-3}$ and exponentially decays into the plasma. A poloidally nonuniform source of neutrals is specified with a peak around the X-point to mimic 2D neutral calculations, such as those in UEDGE. The detailed description of simulations is given in [69]. The animation clearly demonstrates that the turbulence originates inside the separatrix due to the steep density

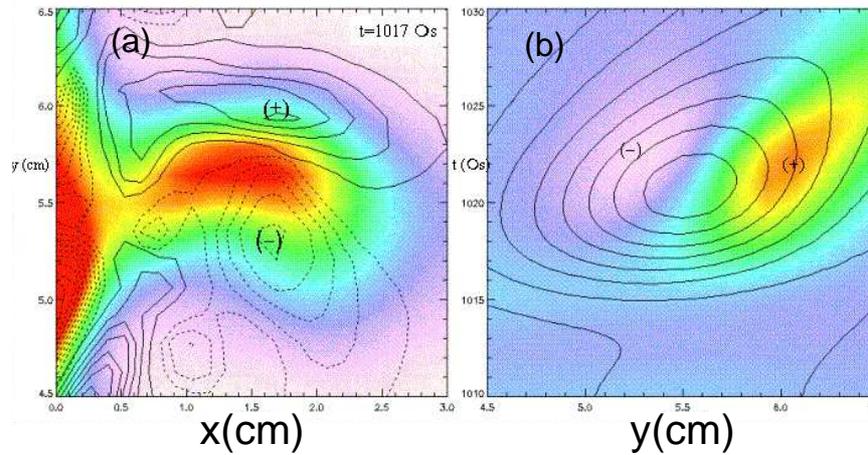


Figure 8: (color online). (a) Blob detached from the separatrix, showing vorticity (contour lines) and density (color); (b) history of blob vorticity at the wall, density (contour lines) and vorticity (color).

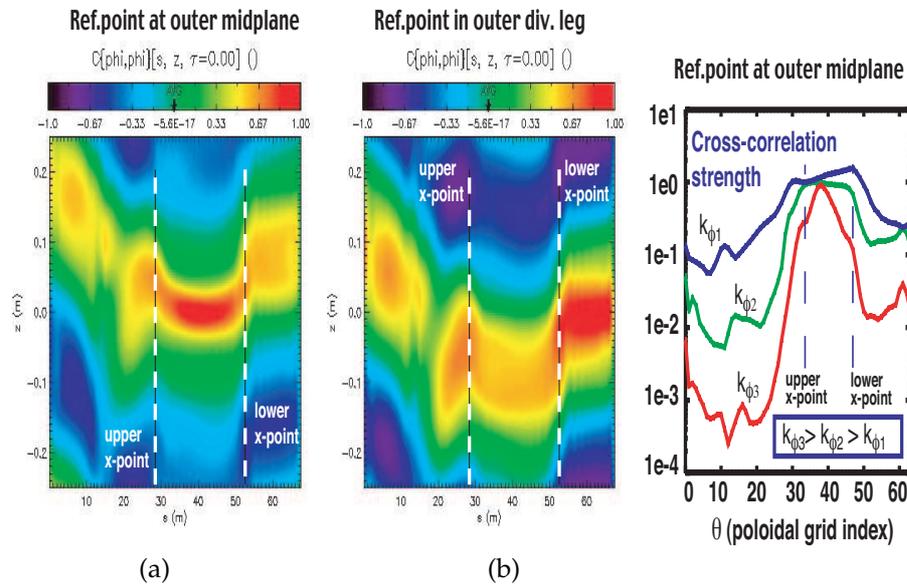


Figure 9: (color online). (a) correlation function for reference point at outer midplane vs poloidal and parallel correlation length; (b) correlation function for reference point at outer divertor leg.

gradient. As density rises, the fluctuating density increases, the large-scale radial mode structure peels off near the separatrix due to poloidal shear flow, and isolated plasma blobs are therefore born. Plasma jets occasionally develop and remain connected to hot core plasma inside the separatrix.

The important properties include [70]: (1) Blob detachment from the separatrix: spatially localized and non-diffusive transport of positive density fluctuations radially outward, as shown in Fig. 8(a). (2) Blob translation from dipole vorticity with the $\mathbf{E} \times \mathbf{B}$ drift

calculated from potential fluctuations, as shown in Fig. 8(b). The self-consistent E-field of the blob is predominantly a dipole field, increasingly as the blob moves away from the separatrix. The radial velocity shows a weak variation with blob radius, as expected from “disconnected” blob models [22, 67]. (3) Blob rotation (monopole vorticity): observed to decay, probably due to T_e relaxation and/or sheath disconnection. (4) Cross correlation analysis indicates a decorrelation of turbulence between the midplane and the divertor leg due to strong X-point magnetic shear [68]. Fig. 9(a) shows that the cross-correlation has cutoffs near both the lower X-point and the upper X-point regions for reference point at outer midplane, and the cutoff is more pronounced for larger poloidal wavenumber, k_θ . Fig. 9(b) shows that the cross-correlation has cutoffs near the X-point regions for reference point at outer divertor leg. Figs. 9(a) and 9(b) also show that the poloidal correlation length is about 1 cm, and the parallel correlation length is about 20 meters.

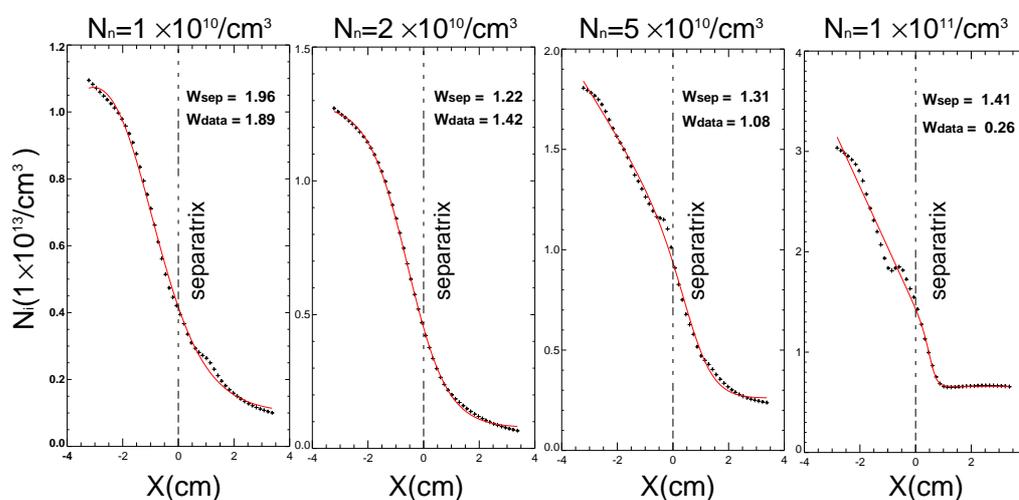


Figure 10: (color online). Background plasma density and its modified tanh fit to the simulation profiles after ~ 0.5 – 1 ms evolution vs. neutral density at the outside midplane.

The simulation results also show the density buildup around the separatrix in L-mode during neutral fueling. The simulation data points with different neutral density at the wall and their fit to a “modified tanh fit” formula [69] are plotted in Fig. 10. The main effect of raising the neutral density (aside from raising the overall density) is to increase the density in the far SOL relative to the top of the density profile. The density gradient scale length parameters are obtained by fitting the modified tanhfit function to the profiles. There is a general overall trend for formation of a “knee” at the base of the profile and for the minimum density gradient scale length W_{data} to decrease with increasing density as observed in the experimental data [71]. However, the center position of the modified tanhfit (“knee” at the base of the profile) is moving toward the SOL and the modified tanhfit is no longer the best fit, due to the appearance of large blob structures as the density increases. The flat density profile in the SOL at high neutral density is a feature of convective transport by localized plasma blobs. The detailed blob dynamics for the case

of neutral density $N_n = 1 \times 10^{11} \text{ cm}^{-3}$ in Fig. 10 is shown in Fig. 8, in the animation, and is analyzed in [22]. It is also found that the density gradient scale length at the separatrix W_{sep} is roughly constant with increasing density.

8 Summary and conclusions

It is shown in this paper that the application of a fluid model is, in many cases, reasonable in the boundary plasma of present fusion devices due to the low temperature and thus high collisionality. A unique boundary plasma turbulence code, BOUT, has been developed that spans the separatrix, including three distinct regions: the outer part of the closed flux region (Edge), the SOL, and the private flux region. The field-aligned coordinates and annular toroidal wedge concept have been used for efficient simulations of the boundary turbulence. The detailed spatial numerical implementation has been presented. The innovative implicit Newton-Krylov iterative method is utilized via a fully implicit solver: PVODE. In order to simulate real experiments, BOUT is designed to couple to the edge plasma transport code UEDGE, and MHD equilibrium codes EFIT/Corsica to get the realistic X-point divertor magnetic geometry and plasma profiles. The BOUT turbulence code suite offers unique and leading-edge numerical and computational resources that enable physical understanding for discovery, design and analysis.

We show that, in examples of BOUT simulation results, as density rises, the fluctuations change from resistive X-point mode to resistive ballooning mode dominated, and from small scale turbulence to large blobs. In the large blob regime at high density, the enhanced radial transport as shown can lead to rapid edge cooling, which leads to a density limit. The description given here is consistent with recent experiments on C-Mod [60,61] and analytical analysis including perpendicular heat convection based on the blob heat transport model [62]. BOUT simulations show that X-point effects can isolate blobs in the main SOL from divertor legs and also provide evidence of instability and fluctuations in divertor legs that is uncorrelated with activity in the main SOL. In summary, our results shed light on the qualitative trends and scalings, and provide suggestions of possible experimental control techniques.

Acknowledgments

This work was performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The research reported here has been partly supported by the Guangbiao Foundation of Zhejiang University. We thank Drs. A. C. Hindmarsh, S. L. Lee, and A. Taylor for the PVODE support, J. A. Boedo, R. H. Bulmer, K. H. Burrell, P. J. Catto, R. H. Cohen, A. M. Dimits, P. H. Diamond, M. Greenwald, R. J. Groebner, S. I. Krasheninnikov, L. L. Lodestro, G. R. McKee, J. R. Myra, W. H. Meyer, W. M. Nevins, R. A. Moyer, L. D. Pearlstein, M. Porkolab, G. D. Porter, M. E. Rensink, T. D. Rognlien, M. N. Rosenbluth, D. A. Russel,

D. D. Ryutov, R. Schneider, B. Scott, A. N. Simakov, E. J. Synakowski, and S. J. Zweben for fruitful physics discussions.

Appendix: Differential operators

A.1 Derivative along unperturbed magnetic field $\vec{B}_0 \cdot \vec{\nabla}$

$$\vec{B}_0 \cdot \vec{\nabla} A = (\vec{\nabla}_z \times \vec{\nabla}_x) \cdot (\vec{\nabla}_y \frac{\partial A}{\partial y}) = \frac{1}{J} \frac{\partial A}{\partial y}. \quad (\text{A.1})$$

A.2 Laplacian ∇_{\perp}^2

$$\nabla^2 A = \frac{1}{J} \sum_i \frac{\partial}{\partial x_i} \left[J \left(\sum_j \frac{\partial A}{\partial x_j} \vec{\nabla}_{x_j} \right) \cdot \vec{\nabla}_{x_i} \right]. \quad (\text{A.2})$$

From Eq. (3.3), neglecting $\partial/\partial y$ terms, also dropping terms with first derivatives $\partial A/\partial x$ and $\partial A/\partial z$ that are small compared to those with second derivatives,

$$\nabla_{\perp}^2 A = |\vec{\nabla}_z|^2 \frac{\partial^2 A}{\partial z^2} + 2(\vec{\nabla}_z \cdot \vec{\nabla}_x) \frac{\partial^2 A}{\partial z \partial x} + |\vec{\nabla}_x|^2 \frac{\partial^2 A}{\partial x^2}. \quad (\text{A.3})$$

Then we obtain

$$\nabla_{\perp}^2 A = (RB_{\theta})^2 \left\{ \frac{\partial^2 A}{\partial x^2} - 2I \frac{\partial^2 A}{\partial z \partial x} + \left[I^2 + \frac{B^2}{(RB_{\theta})^4} \right] \frac{\partial^2 A}{\partial z^2} \right\}. \quad (\text{A.4})$$

A.3 Operator $\vec{B} \times \vec{\nabla} \phi \cdot \vec{\nabla} A$

$$V_E \cdot \vec{\nabla} A = \frac{c}{B^2} \vec{B} \times \vec{\nabla} \phi \cdot \vec{\nabla} A = c \left(\frac{\partial \phi}{\partial z} \frac{\partial A}{\partial x} - \frac{\partial \phi}{\partial x} \frac{\partial A}{\partial z} \right). \quad (\text{A.5})$$

A.4 Operator $\vec{b} \cdot \vec{\nabla} G$

$$\vec{b} \cdot \vec{\nabla} G = \vec{\nabla} A_{\parallel} \times \vec{B} \cdot \vec{\nabla} G = \frac{\partial A_{\parallel}}{\partial x} \frac{\partial G}{\partial z} - \frac{\partial A_{\parallel}}{\partial z} \frac{\partial G}{\partial x}. \quad (\text{A.6})$$

References

- [1] M. Greenwald, J. J. Terry, et al., Nucl. Fusion 28 (1988) 2199; M. Greenwald, Plasma Phys. Control. Fusion 44 (2002) R27-R80.
- [2] S. J. Zweben, J. A. Boedo, O. Grulke, C. Hidalgo, B. LaBombard, R. J. Maqueda, P. Scarin, J. L. Terry, Plasma Phys. Control. Fusion 49 (2007) S1-S23.

- [3] S. I. Krasheninnikov, D. A. D'Ippolito, J. R. Myra, J. Plasma Phys., in progress (2008).
- [4] X. Q. Xu, R. H. Cohen, T. D. Rognlien, J. R. Myra, Phys. Plasmas 7 (2000) 1951.
- [5] B. N. Rogers, J. F. Drake, Phys. Rev. Lett. 81 (1998) 4396.
- [6] B. Scott, Phys. Plasmas 5, 1845 (2001); Plasma Phys. Control. Fusion 49 (2007) S25-S41 and references therein.
- [7] K. Hallatschek, Plasma Phys. Control. Fusion 49 (2007) B137-B148 and references therein.
- [8] W. E. Han, A. Thyagaraja, S. J. Fielding, M. Valovic, Plasma Phys. Control. Fusion 42 (2000) 181-201.
- [9] P. Beyer, S. Benkadda, G. Fuhr-Chaudier, X. Garbet, Ph Ghendrih, Y. Sarazin, Plasma Phys. Control. Fusion 49 (2007) 507-523 and references therein.
- [10] V. Naulin, J. Nucl. Mater. 363-365 (2007) 24-31; V. Naulin, T. Windisch, O. Grulke, Phys. Plasmas 15 (2008) 012307 and references therein.
- [11] M. Yagi, S.-I. Itoh, K. Itoh, P. H. Diamond, Contr. Plasma Phys. 48 (2008) 13-22.
- [12] X. Q. Xu, R. H. Cohen, Contr. Plasma Phys. 38 (1998) 158.
- [13] G. D. Byrne, A. C. Hindmarsh, Int. J. High Perf. Comput. Appl. 13 (1999) 354-365.
- [14] W. D. Gropp, E. Lusk, A. Skjellum, Using MPI Portable Parallel Programming with the Message-Passing Interface, The MIT Press, Cambridge, MA, 1994.
- [15] T. D. Rognlien, D. D. Ryutov, N. Mattor, G. D. Porter, Phys. Plasmas 6 (1999) 1851.
- [16] L. L. Lao, et al., Nucl. Fusion 25 (1985) 1611.
- [17] L. D. Pearlstein, et al., Proceedings 28th EPS Conf. Controlled Fusion Plasma Phys, Madeira, Portugal, June 2001, (EPS, Funchal, 2001), Vol. 25A (ECA), p. 1901; <http://www.cfn.ist.utl.pt/EPS2001/fin/authors/nav/AutP01fr.html>.
- [18] J. R. Myra, D. A. D'Ippolito, X. Q. Xu, R. H. Cohen, Phys. Plasmas 7 (2000) 4622.
- [19] X. Q. Xu, R. H. Cohen, W. M. Nevins, et al., Nucl. Fusion 42 (2002) 21-27.
- [20] A. Mazurenko, M. Porkolab, D. Mossessian, J. A. Snipes, X. Q. Xu, W. M. Nevins, Phys. Rev. Lett. 89 (2002) 225004.
- [21] G. R. McKee, R. J. Fonck, et al., Phys. Plasma 10 (2003) 1712-1719.
- [22] D. A. Russell, D. A. D'Ippolito, J. R. Myra, W. M. Nevins, X. Q. Xu, Phys. Rev. Lett. 93 (2004) 265001.
- [23] D. A. Russell, J. R. Myra, D. A. D'Ippolito, Phys. Plasmas 14 (2007) 102307.
- [24] M. V. Umansky, T. D. Rognlien, X. Q. Xu, R. H. Cohen, W. M. Nevins, Contr. Plasma Phys. 44 (2004) 182-187.
- [25] R. H. Cohen, et al., Nucl. Fusion 47 (2007) 612-625.
- [26] P. B. Snyder, H. R. Wilson, X. Q. Xu, Phys. Plasmas 12 (2005) 056115.
- [27] M. E. Fenstermacher, T. H. Osborne, A. W. Leonard, P. B. Snyder, et al, Nucl. Fusion 45 (2005) 1493-1502.
- [28] J. H. Yu, J. A. Boedo, E. M. Hollmann, R. A. Moyer, D. L. Rudakov, P. B. Snyder, Phys. Plasmas 15 (2008) 032504.
- [29] M. V. Umansky, R. H. Cohen, L. L. LoDestro, X. Q. Xu, Contr. Plasma Phys. 48 (2008) 2731.
- [30] B. Dudson, The 35th European Physical Society Conference on Plasma Physics, 9-13 June 2008 (35th EPS), Hersonissos, Crete, Greece; B. D. Dudson, H. R. Wilson, M. V. Umansky, X. Q. Xu, P. B. Snyder, BOUT++: A framework for parallel plasma fluid simulations, Comput. Phys. Commun., submitted.
- [31] T. D. Rognlien, M. V. Umansky, X. Q. Xu, R. H. Cohen, L. L. LoDestro, J. Nucl. Mater. 337-339 (2005) 327-331.
- [32] S. I. Braginskii, Transport properties in a plasma, in: M. A. Leontovich (Ed), Rev. Plasma Phys., Consultants Bureau, 1 (1965) 205-311.

- [33] A. B. Mikhailovskii, V. S. Tsypin, *Beitr. Plasmaphys.* 24 (1984) 335 and references therein.
- [34] A. N. Simakov, P. J. Catto, *Phys. Plasmas* 10 (2003) 4744; Erratum: A. N. Simakov, P. J. Catto, *Phys. Plasmas* 11 (2004) 2326.
- [35] A. Zeiler, J. F. Drake, B. N. Rogers, *Phys. Plasmas* 4 (1997) 2134.
- [36] A. Zeiler, D. Biskamp, B. N. Rogers, J. F. Drake, *Phys. Plasmas* 5 (1998) 2654.
- [37] B. N. Rogers, J. F. Drake, *Phys. Plasmas* 6 (1999) 2797.
- [38] X. Q. Xu, W. M. Nevins, R. H. Cohen, J. R. Myra, P. B. Snyder, *New J. Phys.* 4 (2002) 53.1-53.15.
- [39] P. C. Stangeby, *The Plasma Boundary of Magnetic Fusion Devices*, Institute of Physics Publishing, Bristol and Philadelphia, 2000.
- [40] A. M. Dimits, *Phys. Rev. E* 48 (1993) 4070.
- [41] M. A. Beer, S. C. Cowley, G. W. Hammet, *Phys. Plasmas* 2 (1995) 2687.
- [42] B. Scott, *Phys. Plasmas* 8 (2001) 447.
- [43] R. D. Falgout, J. E. Jones, U. M. Yang, The design and implementation of hypre, a library of parallel high performance preconditioners, in: A.M. Bruaset, A. Tveito, (Eds), *Numerical Solution of Partial Differential Equations on Parallel Computers*, Springer-Verlag, V. 51 (2006) 267-294.
- [44] J. L. V. Lewandowski, L. E. Zakharov, *Commun. Comput. Phys* 2 (2007) 684.
- [45] M. F. Adams, Y. Nishimura, *Commun. Comput. Phys.* 2 (2007) 881.
- [46] M. Abramowitz, I. A. Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, Applied Mathematics Series-55, Tenth Printing, December 1972.
- [47] P. N. Brown, A. C. Hindmarsh, *J. Appl. Math. Comp.* 31 (1989) 40.
- [48] Y. Saad, *Num. Lin. Alg. Applic.* 1 (1994) 387.
- [49] T. D. Rognlien, X. Q. Xu, A. C. Hindmarsh, *J. Comput. Phys.* 175 (2002) 249-268.
- [50] W. M. Nevins, X. Q. Xu, T. N. Carlstrom, et al., *Proceedings of the 19th International Conference on Fusion Energy*, Lyon, France, 14-19 October 2002, IAEA-CN-94/THP3/07.
- [51] P. B. Snyder, H. R. Wilson, et al., *Phys. Plasmas* 9 (2002) 2037; H. R. Wilson, P. B. Snyder, et al., *Phys. Plasmas* 9 (2002) 1277.
- [52] J. R. Myra, M. Umansky, *The 35th European Physical Society Conference on Plasma Physics*, 9-13 June 2008 (35th EPS), Hersonissos, Crete, Greece, Lodestar Report #LRC-08-120, April, 2008, <http://www.lodestar.com/LRCreports/Linear%20Analysis%20LRC%20Report.pdf>.
- [53] M. V. Umansky, X. Q. Xu, B. Dudson, L. L. Lodestro, *BOU code manual*, LLNL, June, 2006. Available from <http://www.mfescience.org/bout/>. *Comput. Phys. Commun.*, 2008, submitted.
- [54] S. Schlesinger, *Simulation* 32 (1979) 103.
- [55] M. Greenwald, *Computer Phys. Commun.* 164 (2004) 1; P. W. Terry, M. Greenwald, J.-N. Leboeuf, et al., *Phys. Plasma* 15 (2008) 062503.
- [56] P. J. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, New Mexico, 1998; P. J. Roache, *Fundamentals of Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, 1998.
- [57] P. J. Roache, *J. Fluid. Eng.*, *Transactions of the ASME* 124 (2002) 4-10.
- [58] W. L. Oberkampf and T. G. Trucano, *Prog. Aersp. Sci.* 38 (2002) 209; W. L. Oberkampf and M. F. Barone, *J. Comp. Phys.* 217 (2006) 5.
- [59] X. Q. Xu, W. M. Nevins, et al., *Phys. Plasmas* 10 (2003) 1773.
- [60] B. LaBombard, R. L. Boivin, M. Greenwald, et al., *Phys. Plasmas* 8 (2001) 2107; B. LaBombard, J. W. Hughes, D. Mossessian, M. Greenwald, B. Lipshultz, J. L. Terry, et al., *Nucl. Fusion* 45 (2005) 1658.
- [61] J. L. Terry, S. J. Zweben, et al., *Phys. Plasmas* 10 (2003) 1739; T. L. Jerry, J. L. Terry, N. P. Basse,

- I. Cziegler, et al., Nucl. Fusion 45 (2005) 1321.
- [62] D. A. D'Ippolito, J. R. Myra, Phys. Plasmas 13 (2006) 062503.
- [63] T. W. Petrie, A. G. Kellman, et al., Nucl. Fusion 33 (1993) 929.
- [64] S. I. Krasheninnikov, Phys. Lett. A 283 (2001) 368.
- [65] J. R. Myra, D. A. D'Ippolito, D. P. Stotler, S. J. Zweben, B. P. LeBlanc, J. E. Menard, R. J. Maqueda, J. Boedo, Phys. Plasmas 13 (2006) 092509.
- [66] Y. Sarazin, P. Ghendrih, Phys. Plasmas 5 (1998) 4214.
- [67] D. A. D'Ippolito, J. R. Myra, D. A. Russell, et al., 20th IAEA Fusion Energy Conference (Vilamoura, Portugal, 1-6 November 2004), IAEA-CN-116/TH/P6-2.
- [68] M. V. Umansky, T. D. Rognlien, X. Q. Xu, J. Nucl. Mater. 337-339 (2004) 266-270.
- [69] X. Q. Xu, W. M. Nevins, et al., Contr. Plasma Phys. 44 (2004) 105.
- [70] X. Q. Xu, R. H. Cohen, W. M. Nevins, T. D. Rognlien, D. D. Ryutov, M. V. Umansky, L. D. Pearlstein, R. H. Bulmer, D. A. Russell, J. R. Myra, D. A. D'Ippolito, M. Greenwald, P. B. Snyder, M. A. Mahdavi, 20th IAEA Fusion Energy Conference Vilamoura, Portugal, 1-6 November 2004, IAEA-CN-116/TH/1-5, (2004).
<http://www-pub.iaea.org/MTCD/Meetings/PDFplus/fusion-20-preprints/TH1-5.pdf>.
- [71] R. J. Groebner, et al., 19th IAEA FEC (Lyon, 2002) IAEA-CN-94/EX/C2-3.