Extending BOUT++ for solution of edge plasma equations applicable to whole device simulation of tokamaks

Ammar H. Hakim (PI)
Tech-X Corporation
5621 Arapahoe Ave., Suite A
Boulder, CO 80303

Date: October 20th 2011.
Project Performace Period: 06/19/10 to 07/31/11

Grant Number: DE-SC0004435
"Grant supported by DOE Office of Fusion Energy Sciences"

Contents

1 Executive Summary

1.1 Identification and significance of the problem or opportunity

This project proposed to develop improved simulation capabilities for tokamak edge plasmas. In particular, successful completion of the project would have enabled fusion researchers to simulate (1) the warm plasma region adjacent to the tokamak wall known as the scrape-off layer (SOL) where magnetic field lines directly contact material structures together with the associated plasma-wall interactions and (2) the adjacent hotter plasma region know as the pedestal, which is the outermost part of the hot confining core region.

1.2 Background information

1.2.1 Physics of the edge and its importance to tokamak performance

High-performance tokamak plasma discharges (so-called "H-mode" plasmas) are characterized by steep gradients of temperature and density near the plasma separatrix where open field lines are separate from closed field lines. This region is referred to as the "H-mode pedestal" and it serves as a boundary condition to the core regions. Experiments and core transport simulations indicate that core profiles are relatively insensitive to the overall values of density and temperature, but that the shape is set by stiff transport due to kinetic turbulence. Thus, the overall fusion gain depends sensitively on the temperature and density values at the top of the pedestal [3] by acting as the boundary condition to the core plasma region. The physics determining the pedestal is not fully understood, and is the subject of active investigations [4].

The difficulty of understanding the physics of tokamak edge plasmas as compared to core plasmas is due to the lack of separation of time scales as well as the additional physics that cooler plasmas bring. Explanations of the pedestal width and height include the combination of kinetic ballooning modes and peeling-ballooning mode instabilities in the EPED1 model [5], neoclassical transport due to ion orbit loss [6] and neutral sourcing [7, 8]. Although the EPED1 model has had the most success in validation studies [4], it is difficult to apply to results such as the ELM free period of NSTX when the lithium divertor is used [9]. That is, although the EPED1 model is used to explain the pedestal limits, it cannot explain how the plasma evolves to these these limits.

Among the many improvements needed in edge modeling is the ability to include the physics of peeling-ballooning modes and kinetic ballooning modes in a way that is self-consistent the modeling of the transport needed to reach those limits. Because the time scale between ELMs can be as fast as a 100 msec time scale, which is still long compared to the faster kinetic or instability time scales, integration of the turbulent and instability processes within an integrated model is needed. Because fluid models are used to understand the peeling-ballooning limit [10], kinetic ballooning modes [11], and comprehensive edge transport on long time scales [12], integration of this wide variety of time scales within a fluid model is a logical next step. This is not to imply that we view fluid models as sufficient, but rather that the numerical and validation studies in this work represent an advance in the state of the art in a tractable manner.

In the core region, the understanding of the impact of turbulence on transport time scales has been greatly facilitated by the availability of the "1.5 D" transport model as well as the development of reduced models using the quasilinear approximation [13, 14, 15]. The logical extension of this method is the embedded turbulence technique [16, 17] which uses a gyrokinetic code to calculate the turbulence fluxes at each radial surface, and then achieves self-consistency between the turbulence and profiles using a Newton iteration. Although there have been discussion of the effects of non-locality in the core region (discussed further below), the success of the quasilinear, reduced models [3, 14, 18] gives confidence that in the core plasma at least, the embedded turbulence model can be used to explain many features of core transport.

The integration of turbulent fluxes on the transport time scale in the edge region is considerably more complicated due to the nature of the turbulence in the edge. Experimentally it has been observed that the plasma contains intermittent, highly localized structures that carry much of the density and energy out of the plasma [19]. This intermittent transport has been observed in fluid transport simulations as well [20]. Other non-local effects have been observed in edge kinetic simulations [21] implying that understanding non-local effects on transport time scales is a ubiquitous issue. Techniques for using turbulence simulations that have intermittent transport to advance transport equations long time scales were investigated by Shestakov and colleagues [22]. In their work, the transport is advanced based on fluxes from turbulence calculation. While the calculation was based on using the Hasegawa-Wakatani equations, proof-of-principle calculation demonstrates that the integrated simulation takes less time than just running the turbulence code alone.

The ability to take this proof-of-concept simulation to full production simulation is complicated by the complexity of the edge simulations even in the fluid limit. While similar in spirit to the core transport codes in that transport fluxes need to be obtained from a type of sub-grid model, edge models must also deal with higher collisionality, the presence of a significant neutral gas component owing to ion recycling and sputtering from adjacent surfaces, large non-coronal (equilibrium) line-radiation energy loss from hydrogenic and impurity species, large finite ion-gyro-orbit effects in steep radial gradients, and regimes of large-amplitude, intermittent turbulence. Edge codes include many of these aspects in an integrated way, but use varying degrees of approximation. A detailed discussion of many of the physics processes in the edge is given by Stangeby[23], including plasma-wall interactions. Two key edge issues for large devices are how to build a high-temperature edge that has a strong positive influence on core energy confinement, and distributing the escaping plasma power to prevent excessive erosion or melting of material surfaces.

1.2.2 Fluid modeling of the edge using UEDGE

The most complete 2D edge fluid transport code developed in the U.S. is UEDGE [1, 12], which includes full single- or double-null X-point geometry with a simulation domain including the region spanning well inside the separatrix and extending to the outer wall and the divertor plate region. Neutrals and plasma-wall interactions are very important in the edge. Fluid neutral models [12] are fast, but calculation of neutral transport in the low density edge regions, the hot core interface, and charge-exchange fluxes to walls typically requires a kinetic model. For this, the 3D DEGAS-2 Monte Carlo codes [24] has been integrated with

UEDGE on a limited basis. For wall recycling of ions into neutrals, UEDGE currently uses a simple static model.

UEDGE is time-dependent and unique among present edge transport codes in utilizing a fully-implicit preconditioned Newton-Krylov method with the neutral component included at the same implicit level as the plasma. Owing to the large range of neutral and plasma timescales, this implicit package allows for efficient determination of edge transport equilibria. The result is that UEDGE spends between 30-50 percent of overall run-time on (non)linear Newton-Krylov solves for problems of size (80×40), with another 30% going to forming and inverting (partially with ILUT) the full finite-difference Jacobian. The full finite-difference Jacobian is needed due to the strong coupling of many nonlinear edge physics components. The remaining time in UEDGE is devoted to physics-equation evaluations.

The basic aspects of the edge plasma model can be understood by considering the equations of continuity, momentum, and energy for both the electrons and ions as given by Braginskii [25]. The continuity equations have the form

$$\frac{\partial n_{i,e}}{\partial t} + \nabla \cdot (n_{i,e}\mathbf{v}_{i,e}) = S^p_{i,e} \tag{1}$$

where $n_{i,e}$ and $\mathbf{v}_{i,e}$ are the ion and electron densities and mean velocities respectively. The source term $S^p_{i,e}$ arises from ionization of neutral gas and recombination. The momentum equations are given by

$$\begin{aligned}\frac{\partial n_{i,e}m_{i,e}\mathbf{v}_{i,e}}{\partial t} &+ \nabla \cdot (n_{i,e}m_{i,e}\mathbf{v}_{i,e}\mathbf{v}_{i,e}) = \\ &- \nabla P_{i,e} \pm e n_{i,e}(\mathbf{E} + \mathbf{v}_{i,e} \times \mathbf{B}) - \mathbf{F}_{i,e} - \mathbf{R}_{i,e} + \mathbf{S}^m_{i,e}.\end{aligned}$$

Here m_i is the ion mass, $P_{i,e} = n_{i,j}T_{i,e}$ is the pressure with $T_{i,e}$ being the temperature, $\pm e$ is the ion and electron charges, $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic field, $\mathbf{F}_{i,e} = \nabla \cdot \mathbf{\Pi}_{i,e}$ is the viscous force, and $\mathbf{R}_{i,e}$ is the friction force. The source $\mathbf{S}_{i,e}$ contains effects of external momentum exchange, e.g. with neutrals.

The ion and electron energy equations can be written as

$$\frac{\partial}{\partial t}\left(\frac{3}{2}P_{i,e}\right) + \nabla \cdot \left(\frac{5}{2}\mathbf{v}_{i,e}P_{i,e} + \mathbf{q}_{i,e}\right) = \mathbf{v}_{i,e}\nabla \cdot P_{i,e} - \mathbf{\Pi}_{i,e} \cdot \nabla\mathbf{v}_{i,e} + Q_{i,e} + S^T_{i,e}. \tag{2}$$

Here $\mathbf{q}_{i,e}$ are the heat fluxes, and $Q_{i,e}$ are volume heating terms including Joule heating. The final term, $S^T_{i,e}$, are the energy sink and sources due to atomic and radiation processes. Since anomalous radial transport radial transport processes are believed important they are included by allowing enhanced ion and electron collisions frequencies in the classical model.

UEDGE solves a reduced set of the Braginskii transport equations suitable for long time scale transport simulations. The equations are written in a form that allows for identification of the physics terms and the option of turning on and off terms to allow better physics understanding. The specific form of the equations are found in Ref. [12].

For the neutral equations, the simplest model, and the one we have implemented in the Phase I project, is:

$$\frac{\partial}{\partial t}n_n + \nabla \cdot (n_n\mathbf{v}_n) = (\langle\sigma_r v_e\rangle - \langle\sigma_i v_e\rangle)\, n_n n_e \tag{3}$$

where $\sigma_{r,i}$ are the recombination and ionization cross-sections respectively. The neutral velocity is calculated using a simple drift model:

$$\mathbf{v}_n = \frac{\nabla\left(n_n T_n\right)}{m_i n_n \left(n_i \left\langle \sigma_{cx} v_i \right\rangle + n_e \left\langle \sigma_i v_e \right\rangle\right)} \tag{4}$$

where σ_{cx} is the charge-exchange cross-section. Obtaining the atomic physics information is an important part of the modeling and is discussed in the tasks.

For the spatial discretization of these equations, UEDGE uses a curvilinear multiblock grid shown in Fig. 1. The equations are discretized using a finite-volume scheme. The advection terms are discretized using a first-order upwind scheme and the other terms are discretized using central differences. This reduces the overall order of the scheme to first-order, specially when cross-field diffusion terms are turned on and the advection terms are dominant.

1.2.3 Overview of BOUT++

BOUT++ [26] is a new, highly-adaptable, object-oriented C++ code for performing parallel plasma fluid simulations with an arbitrary number of equations in 3D curvilinear coordinates using finite-differences. It is based on the original BOUT 3D 2-fluid tokamak edge simulation code [27], borrowing ideas and algorithms. However, BOUT++ has been significantly altered and extended to simulate tokamak edge plasmas flexibly and efficiently. The methods used are very general: any coordinate system metric tensor $g^{ij} = g^{ij}(x, y)$ can be specified, allowing the code to be used to simulate plasmas in slab, sheared slab, cylindrical and non-orthogonal coordinate systems such as flux coordinates for tokamak simulations.

BOUT++ uses modern software engineering techniques to automate the common tasks needed for fluid finite-difference simulation code, and to separate details such as differential geometry, parallel communication, and file I/O from the equations to be solved. The physics equations being solved are clearly exposed in one place, and can be easily changed with minimal knowledge of the inner workings of the code. This allows the user to focus on the physics, rather than programming.

The BOUT++ code can be separated into the following parts:

- Time integration using the Sundials CVODE package,
- I/O using the NetCDF format,
- Low-level data handling,
- Parallel communication using MPI,
- Finite-differencing schemes, and
- List of physics modules determining the equations to be solved.

The work of actually solving the equations is done using the time-integration solver CVODE. This is used as a “black-box”, requiring no information about the equations themselves, but simply the values of the tendencies. The equations to be solved are written in

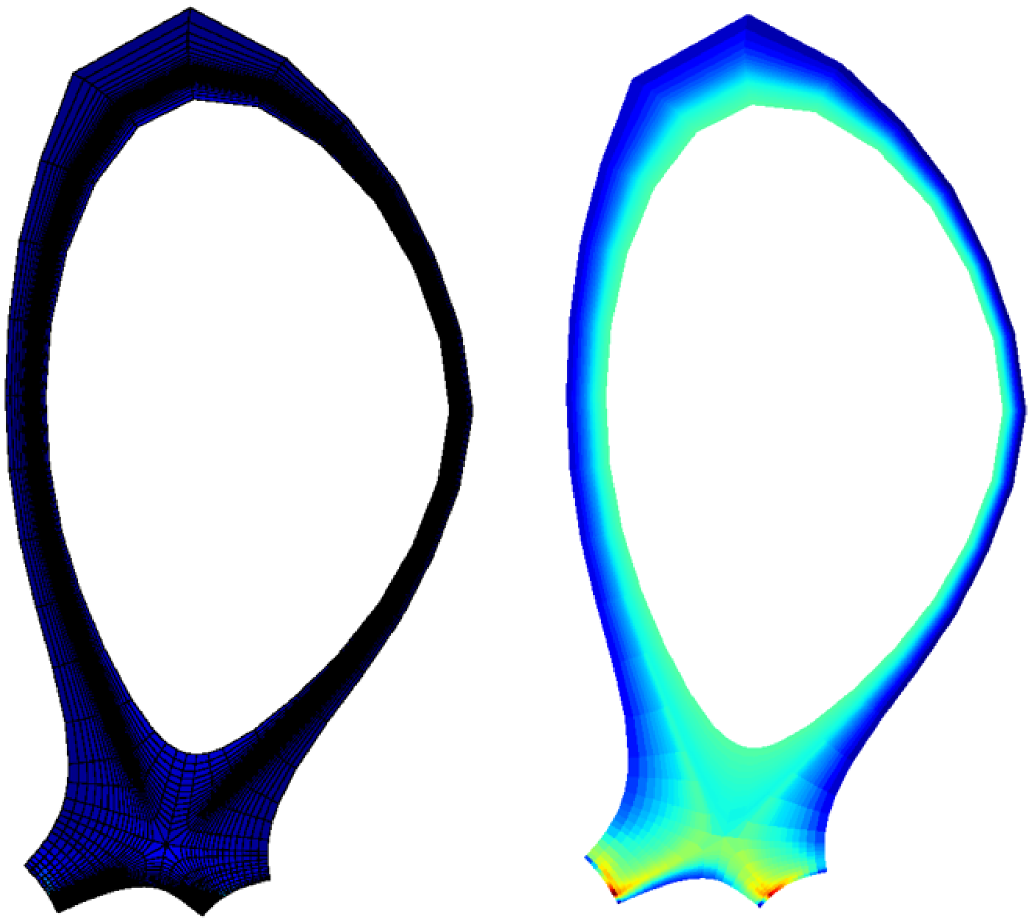

Figure 1: UEDGE grid (left) and solution (right) for DIII-D tokamak edge simulation. The ion temperature is shown. The UEDGE grid consists of three sub-grids: core region just inside the separatrix, SOL region and private flux region. Each of these regions is represented by a rectangular grid in the computational space. The grid transforms are computed such that one set of grid lines lie along the equilibrium magnetic field and other set is orthogonal to it. Multiblock grids allow for simple coarse/refine operations as these are performed in computational space.

semi-discrete form $\partial f/\partial t = \mathbf{L}(f)$, where f is a vector of unknowns and $\mathbf{L}(f)$ is the operator containing spatial-derivatives and the sources in the equations. The operator is computed by BOUT++ and passed to CVODE, which, in turn, calls BOUT++ with a new state f which is then supplied $\mathbf{L}(f)$.

Parameters to BOUT++ consist of text input file with `name = value` pairs and a binary grid file. The text file describes the simulation parameters, while the binary file describes the geometry of the domain.

BOUT++ makes extensive use of C++ operator overloading to abstract mathematical operations. For example,

```
Scalar3D a,b,c;
real r;
a=b+c;

Vector2D x,y,z;
x=y*a; // multiply by scalar field
a=x*y; // dot product
x=y^z; // cross product
```

Using advanced C++ techniques, the temporaries inherent to a naive implementation of operator overloading are avoided. This notation allows the basic equations to be expressed a natural manner. In addition to operator overloading of algebraic equations, BOUT++ also provides a notation for expressing differential operators in a succinct and simple manner. For example, the mathematical expression $\mathbf{v} = \nabla \times \mathbf{a}$ can be written as

```
Vector2D v, a;
v = Curl(a);
```

Other operators for use in tokamak problems are defined; e.g., `Grad_par` which computes the gradient parallel to the magnetic field or `Div_par` which computes the divergence parallel to the magnetic field. With this notation, the ideal-MHD momentum and magnetic field equations

$$\frac{\partial \mathbf{v}}{\partial t} = -\mathbf{v} \cdot \nabla \mathbf{v} - \frac{1}{n} \nabla p \tag{5}$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times \mathbf{v} \times \mathbf{B} \tag{6}$$

can be written as

```
Vector3D dvdt = -V_dot_Grad(v,v) - Grad(p)/n
Vector3D dBdt = Curl(v ^ B)
```

In the Phase I project we have demonstrated that we can implement edge fluid equations in BOUT++. In particular, as explained in the Phase I results section, we have implemented a general multi-species reaction package for BOUT++, implemented a simple neutral fluid model and also incorporated mesh generation and atomic physics data into stand-alone libraries.

2 Phase I Accomplishements

The Phase I project was funded for 6 months of 1 FTE. In this period we demonstrated the feasibility of implementing edge transport equations in BOUT++ and also developed, along the way, two libraries. The first library incorporates a new mesh generator that take an experimental equilibrium reconstruction and creates the mesh needed for BOUT++. The second library incorporates atomic physics data for hydrogenic species and certain impurities. These data were extracted from UEDGE. The mesh generation library now frees a BOUT++ user from having to run UEDGE to create the mesh for tokamak simulations and, in fact, makes the mesh generation process directly available to the C++ application code.

Progress on each task proposed in the Phase I project is described in brief below. The subsections have the same title as the tasks proposed in the Phase I for convenience.

2.1 Task 1: Build, port and exercise BOUT++ on multiple platforms

In the first task of this project we gained familiarity with the BOUT++ code by building it on multiple platforms. This was not a simple task as there are several requirements to make the code usable in a coupled framework needed in the Phase II project. In particular, the build of the needed I/O library (netCDF) was fixed to allow building it without conflicts with other linked libraries in parallel. We have also started to add a layer to write the data out as HDF5 instead of netCDF using the Vizschema markup system[1]. We will complete this in the Phase II project and this will allow the use of the Visit visualization tool to plot the data from BOUT++.

Although BOUT++ is a finite-difference C++ code it does not perform its own grid generation and, for tokamak edge problems, relies on an external grid generated by UEDGE. Hence, we have developed a new grid generator that allows a user to make the grid directly in the C++ application code. This eliminates the need to have a UEDGE build or run. Initially, we had only devoted 1/2 month of 1 FTE to this whole task as the assumption was that the UEDGE grid generation code would be easily extracted. However, the grid generation took significant effort, and as mentioned before, we have written a new proposal to extend our generator to be useful to NIMROD and unstructured mesh codes like M3D, which can benefit from a multi-block mesh generator that has flux-alignment.

Most experimental equilibrium reconstructions are available as eqdsk files. Hence, a C++ class was written to read both g- and a-eqdsk files. The $\psi(R, Z)$ data from the equilibrium is then fed to a contouring code that determines the contour curve given a ψ value. The contouring code works in two stages: in the first stage the contour is traced using a bi-linear interpolation scheme that loops over each cell determining a possible intersection of the required contour with the cell edges. If an intersection is found a straight segmented is added to the contour curve. One needs to take care that the contour curve is a set of continuous curves and not a collection of possibly disjoint segments (as done in plotting tools). In the second stage the contour is "improved" by creating a bi-cubic or spline interpolation of the equilibrium data. In this improvement stage a non-linear equation is solved to move points

[1]See `https://ice.txcorp.com/trac/vizschema`

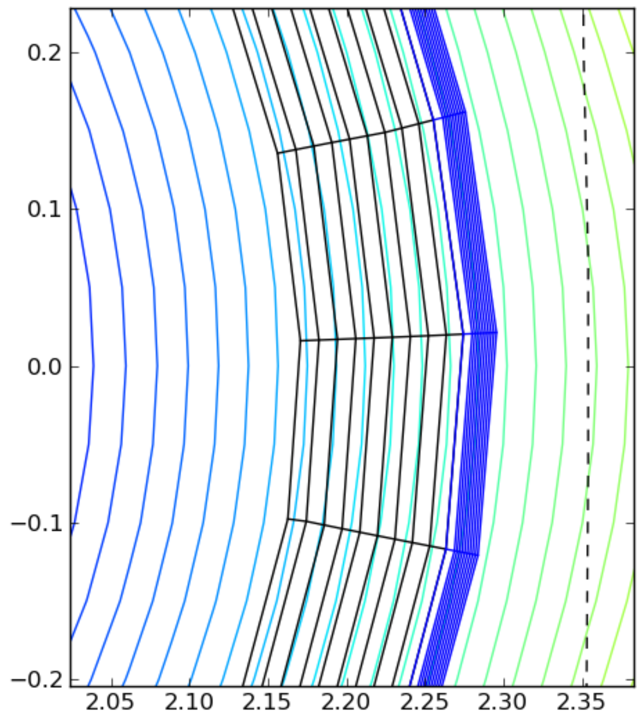

Figure 2: Zoom into the edge grid as it is being generated. The black lines represent the core grid and the blue lines the SOL grid. The contours of $\psi(R, Z)$ are also shown to display the close alignment of the grid with the flux surfaces. The grid packing in this is plot is non-uniform, though not realistic to enable easier visualization. More accurate packing can be controlled by input parameters to the mesh generator.

determined from the bi-linear algorithm to the location on the "correct" curve obtained from the bi-cubic or spline interpolation. The second stage significantly improves the accuracy of the traced contour, specially as an X-point is approached.

The grid generation itself proceeds as follows. First, the separatrix is determined, either read from the eqdsk file or using a simple heuristic based on single or double null geometry. Then, another contour is traced just inside the separatrix. A set of nodes is laid out on the separatrix that serve as the poloidal nodal points. A line is drawn in the $\nabla\psi$ direction from each of these nodes and the intersection point with the inner contour is determined. The intersection segment serves as the cell edge. This procedure is repeated with each core contour, generating a mesh with orthogonal mesh lines. An iterative improvement is needed to ensure exact orthogonality of the generated mesh. A similar procedure is used to generate the SOL mesh. Finally, the nodal mesh points on the "legs" of the separatrix below the X-point are used to generate the private flux region of the mesh. Figs. 2 and 3 show an example of a generated mesh from equilibrium reconstruction from a DIII-D shot.

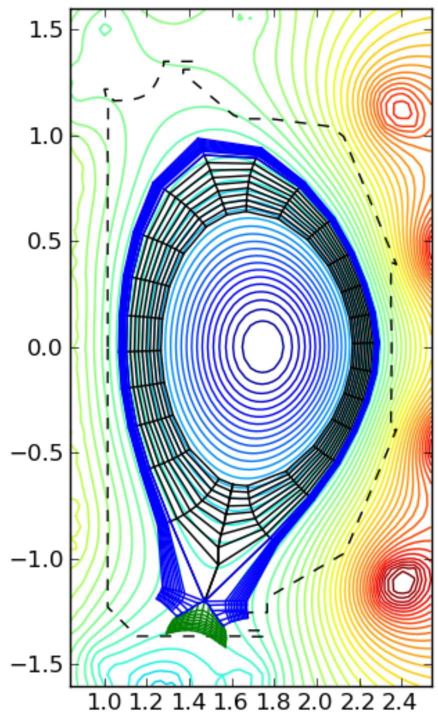

Figure 3: Example of a generated grid for DIII-D shot 110465, 3500 ms into the discharge. The blue lines represents the SOL, the black the core regions and the green the private flux. Note the curving grid lines perpendicular to $\psi(R, Z)$ showing grid orthogonality. Also shown are the tokamak wall and the flux contours. We are presently working on improving the grid around the X-point and in the private flux region.

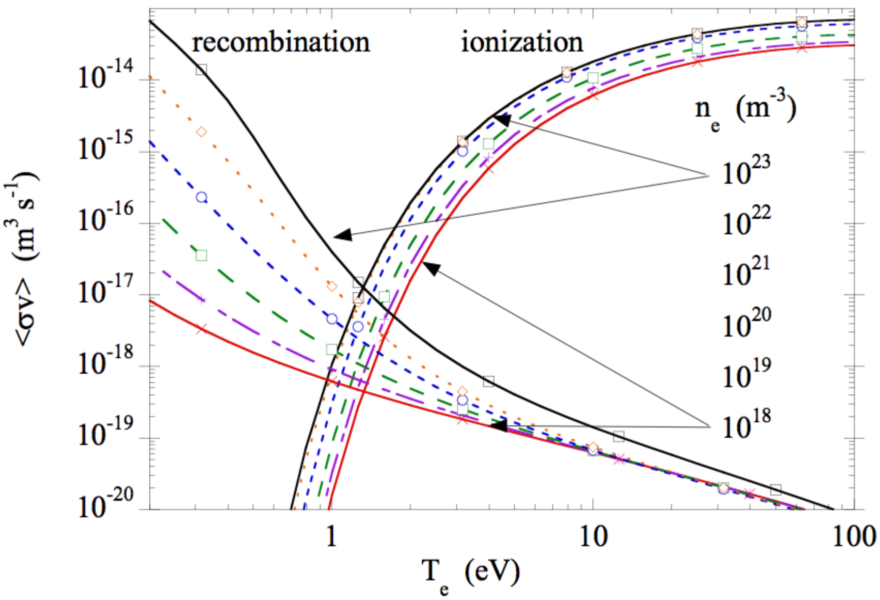

Figure 4: Ionization and recombination rates for various densities as a function of temperature. These data were extracted from UEDGE.

2.2 Task 2: Catalog set of benchmark UEDGE simulations

We have put together a catalog of representative UEDGE simulations. We have created UEDGE input files for both slab and tokamak geometries, with and without density evolution, in order to have a set of varied benchmark problems to compare BOUT++ results. In addition, we have also created input files with drift terms to ensure we have comparison cases with more complete physics. This collection of benchmark cases will be more useful in the Phase II project once the transport equations are fully implemented in BOUT++.

2.3 Task 3: Abstract atomic physics from UEDGE and prepare a library with flexible API to such data

We have abstracted out the atomic physics data from UEDGE and put it into the freely available txphysics library. For the Phase I we decided to only extract the most widely used and recommended data from UEDGE, i.e. data originally obtained from DEGAS2. This data is the one used in most UEDGE calculations we have seen to date.

The particular data sets we extracted were the ionization and recombination rates for hydrogen species and several impurity species. Once the data was copied over to txphysics we wrote a C API to interpolates the data and make it available at arbitrary values of temperature and/or density. A linear interpolation was implemented in log-log space as done in UEDGE. An example of extracted rates is shown in Fig. 4. These data were originally computed using DEGAS2 and are the *effective* ionization and recombination rates, including multi-step ionization and radiative as well as three-body recombination.

An unanticipated benefit, in addition to use in the edge transport solver in BOUT++,

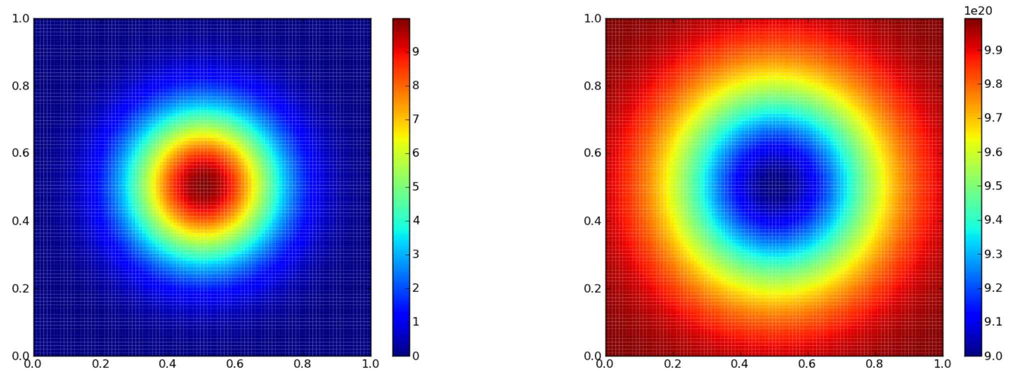

Figure 5: Slab geometry simulation of neutral transport implemented in BOUT++. In this simulation a hydrogen species was used and a fixed ion temperature Gaussian pulse (left panel in eV) was initialized. This causes ionization of the neutrals and an outward diffusion due to the temperature gradient. The right panel shows the density of neutrals which shows depletion of neutrals near the center. Ionization and recombination rates from the data extracted from UEDGE were used in this calculation.

is the use of the hydrogenic ionization and recombination rate data in another DoE funded project to compute capillary discharge problems. In particular, the recombination rates extracted from UEDGE are *effective* rates, including three body recombination as well as radiative recombination and other effects. This allowed us to to a use better model than a simple Arrhenius form for the ionization and recombination in the capillary discharge problem.

The extracted data and implemented interpolation code is freely available via an svn checkout[2] from Tech-X Corporation's subversion server. In the Phase II project, we will provide a website for the project, improve the documentation, and provide tarballs for easier distribution.

2.4 Task 4: Add an initial implementation of UEDGE fluid equations into BOUT++, including a simple neutral fluid model

The basic form of the transport equations we have implemented corresponds to that implemented in the a number of edge transport codes based on classical fluid equations. In particular, we do not solve the complete momentum equation but simply evolve the parallel velocity, in addition to the continuity and electron and ion fluid equations. The ion continuity equation, in the poloidal plane, becomes

$$\frac{\partial n_i}{\partial t} + \frac{1}{V}\frac{\partial}{\partial x}\left(\frac{V}{h_x} n_i u_{ix}\right) + \frac{1}{V}\frac{\partial}{\partial y}\left(\frac{V}{h_y} n_i u_{iy}\right) = (\langle\sigma_i v_e\rangle - \langle\sigma_r v_e\rangle) n_e n_n, \quad (7)$$

where $V = 2\pi R h_x h_y$ is the volume element for toroidal geometry, h_x, h_y are metric terms, n_n is the neutral density, $\langle\sigma_i v_e\rangle$ and $\langle\sigma_r v_e\rangle$ are the rate coefficients for recombination and

[2] `svn co https://ice.txcorp.com/svnrepos/code/txphysics/trunk txphysics`. Access permission is be needed to obtain this code although we plan to make the access anonymous.

ionization, respectively. These rates are shown, for example, in Fig. 4. The velocity $\mathbf{u}_i$ is the ion fluid velocity with the classical cross-field pressure or temperature gradient terms omitted since these have zero divergence, or cancel with gyroviscous terms. The neutral density is determined from the neutral continuity equation, Eq. (3), and the neutral velocity is determined from the simple drift-diffusion model Eq. (4). In this initial implementation we have run BOUT++ with the simple edge model in the slab geometry case.

An example benchmark calculation is shown in Fig. 5. In this calculation we initialized the simulation with a Gaussian temperature profile that peaks to 10eV in the middle of the domain and cools off rapidly as one moves radially outward. We held this temperature fixed and evolved the neutral and plasma density. It is seen that ionization depletes the neutrals in the hot region. Further, the temperature gradient causes a diffusion of the neutrals outwards.

In the Phase II project we will complete the implementation (see Task 2) and run the calculations on a realistic edge grid.

2.5 Task 5: Verify BOUT++ results with UEDGE for collected cases

We have made only preliminary progress on this task. The reason is that the edge transport equations are not completely implemented as the comprehensive physics was planned to be added in the Phase II project. See Task 2 below (section 5.2). To facilitate comparison, we created a set of python scripts to extract data from BOUT++ (most BOUT++ analysis scripts are in IDL) to allow for easy comparison with UEDGE results. For this we have developed interpolation functions to make line-outs along a given ray cutting across the highly skewed edge mesh.

References

[1] T. Rognlien, J. Milovich, M. Rensink, and G. Porter, "A fully implicit, time dependent 2-d fluid code for modeling tokamak edge plasmas," *Journal of Nuclear Materials*, vol. 196–198, pp. 347–351, 1992.

[2] B. D. Dudson, M. V. Umansky, X. Q. Xu, P. B. Snyder, and H. R. Wilson, "Bout++: a framework for parallel plasma fluid simulations," *arXiv*, vol. physics.plasm-ph, Jan 2008.

[3] J. Kinsey, G. Staebler, and R. Waltz, "Burning plasma confinement projections and renormalization of the GLF23 drift-wave transport model," *Fus. Sci. Tech.*, vol. 44, p. 763, 2003.

[4] R. Groebner, A. Leonard, P. Snyder, T. Osborne, C. Maggi, M. Fenstermacher, C. Petty, and L. Owen, "Progress towards a predictive model for pedestal height in diii-d," *Nucl. Fusion*, vol. 49, p. 085037, Jul 2009.

[5] P. B. Snyder, N. Aiba, M. Beurskens, R. J. Groebner, L. D. Horton, A. E. Hubbard, J. W. Hughes, G. T. A. Huysmans, Y. Kamada, A. Kirk, C. Konz, A. W. Leonard, J. Lönnroth, C. F. Maggi, R. Maingi, T. H. Osborne, N. Oyama, A. Pankin, S. Saarelma, G. Saibene, J. L. Terry, H. Urano, and H. R. Wilson, "Pedestal stability comparison and iter pedestal prediction," *Nucl. Fusion*, vol. 49, p. 5035, Aug 2009.

[6] C. S. Chang, S. Ku, and H. Weitzner, "Numerical study of neoclassical plasma pedestal in a tokamak geometry," *Phys. Plasmas*, vol. 11, p. 2649, May 2004.

[7] M. A. Mahdavi, R. Maingi, R. J. Groebner, A. W. Leonard, T. H. Osborne, and G. Porter, "Physics of pedestal density profile formation and its impact on h-mode density limit in burning plasmas," *Physics of Plasmas*, vol. 10, no. 10, pp. 3984–3991, 2003.

[8] R. Groebner, M. Mahdavi, A. Leonard, T. Osborne, G. Porter, R. Colchin, and L. Owen, "The role of neutrals in high-mode (h-mode) pedestal formation," *Phys. Plasmas*, vol. 9, p. 2134, 2002.

[9] H. W. Kugel, M. G. Bell, J.-W. Ahn, J. P. Allain, R. E. Bell, J. A. Boedo, C. E. Bush, D. A. Gates, T. Gray, S. M. Kaye, R. Kaita, B. P. Leblanc, R. Maingi, R. Majeski, D. K. Mansfield, J. E. Menard, D. Mueller, M. Ono, S. F. Paul, R. Raman, A. L. Roquemore, P. W. Ross, S. A. Sabbagh, H. Schneider, C. H. Skinner, V. A. Soukhanovskii, T. Stevenson, J. R. Timberlake, W. R. Wampler, and L. E. Zakharov, "The effect of lithium surface coatings on plasma performance in the national spherical torus experiment," *Phys. Plasmas*, vol. 15, p. 056118, Jan 2008.

[10] P. B. Snyder, H. R. Wilson, and X. Q. Xu, "Progress in the peeling-ballooning model of edge localized modes: Numerical studies of nonlinear dynamics," *Physics of Plasmas*, vol. 12, no. 5, p. 056115, 2005.

[11] P. B. Snyder and G. W. Hammett, "A landau fluid model for electromagnetic plasma microturbulence," *Phys. Plasmas*, vol. 8, p. 3199, Jul 2001.

[12] T. D. Rognlien, D. D. Ryutov, N. Mattor, and G. D. Porter, "Tow-dimensional electric fields and drifts near the magnetic separatrix in divertor tokamaks," *Physics of Plasma*, vol. 6, pp. 1851–1857, May 1999.

[13] R. E. Waltz, G. M. Staebler, W. D. Dorland, G. W. Hammett, M. T. Kotschenreuther, and J. A. Konings, "A gyro-landau-fluid transport model," *Phys. Plasmas*, vol. 4, p. 2482, Jul 1997.

[14] G. Bateman, A. H. Kritz, J. E. Kinsey, and A. J. Redd, "Multi-mode transport modeling of the international thermonuclear experimental reactor (iter)," *Physics of Plasmas*, vol. 5, no. 6, pp. 2355–2362, 1998.

[15] G. M. Staebler, J. E. Kinsey, and R. E. Waltz, "A theory-based transport model with comprehensive physics," *Phys. Plasmas*, vol. 14, p. 055909, Jan 2007.

[16] J. Candy, C. Holland, R. E. Waltz, M. R. Fahey, and E. Belli, "Tokamak profile prediction using direct gyrokinetic and neoclassical simulation," *Phys. Plasmas*, vol. 16, p. 060704, Jan 2009.

[17] M. Barnes, I. Abel, W. Dorland, T. Görler, G. Hammett, and F. Jenko, "Direct multiscale coupling of a transport code to gyrokinetic turbulence codes," *Phys. Plasmas*, vol. 17, p. 056109, 2010.

[18] J. E. Kinsey, G. M. Staebler, and R. E. Waltz, "The first transport code simulations using the trapped gyro-landau-fluid model," *Phys. Plasmas*, vol. 15, p. 055908, Jan 2008.

[19] J. A. Boedo, D. L. Rudakov, E. M. Hollmann, D. S. Gray, K. H. Burrell, R. A. Moyer, G. R. McKee, R. J. Fonck, P. C. Stangeby, T. E. Evans, P. B. Snyder, A. W. Leonard, M. A. Mahdavi, M. J. Schaffer, W. P. West, M. E. Fenstermacher, M. Groth, S. L. Allen, C. J. Lasnier, G. D. Porter, N. S. Wolf, R. J. Colchin, L. Zeng, C. Wang, J. G. Watkins, T. Takahashi, and D.-D. Team, "Edge-localized mode dynamics and transport in the scrape-off layer of the diii-d tokamak," *Phys. Plasmas*, vol. 12, p. 072516, Jan 2005.

[20] G. S. Xu, B. N. Wan, W. Zhang, Q. W. Yang, L. Wang, and Y. Z. Wen, "Multiscale coherent structures in tokamak plasma turbulence," *Phys. Plasmas*, vol. 13, p. 102509, Jan 2006.

[21] C. S. Chang, S.-H. Ku, P. H. Diamond, Z. Lin, S. E. Parker, T. S. Hahm, and N. Samatova, "Compressed ion temperature gradient turbulence in diverted tokamak edge," *Phys. Plasmas*, vol. 16, p. 056108, Jan 2009.

[22] A. Shestakov, R. H. Cohen, J. Crotinger, L. L. Lodestro, A. Tarditi, and X. Xu, "Self-consistent modeling of turbulence and transport," *J. Comp. Phys*, vol. 185, no. 2, pp. 399–426, 2003.

[23] P. C. Stangeby, *The plasma boundary of magnetic fusion devices.* Institute of Physics Pub., 2000.

[24] D. P. Stotler, C. F. F. Karney, M. E. Rensink, and T. D. Rognlien, "Coupling of parallelized degas 2 and uedge code," *Contrib. Plasma Phys.*, vol. 40, p. 221, 2000.

[25] S. I. Braginskii, "Transport Processes in a Plasma," *Reviews of Plasma Physics*, vol. 1, pp. 205–311, 1965.

[26] B. Dudson, M. Umansky, X. Xu, P. Snyder, and H. Wilson, "Bout++: A framework for parallel plasma fluid simulations," *Computer Physics Communications*, vol. 180, no. 9, pp. 1467–1480, 2009.

[27] X. Q. Xu, R. H. Cohen, T. D. Rognlien, and J. R. Myra, "Low-to-high confinement transition simulations in divertor geometry," *Phys. Plasmas*, vol. 7, p. 1951, 2000.

[28] U. D. of Energy, "Facilities for the future of science: A twenty-year outlook." `http://www.er.doe.gov/about/Future/Facilities\for\the\Future\of\Science.htm`, 2003.