Introduction and overview of BOUT++

Ben Dudson

York Plasma Institute, Department of Physics,
University of York, Heslington, York YO10 5DD, UK

BOUT++ Workshop

15^{th} September 2014

Tokamak edge simulations

- Transition region between hot core and material surfaces
- Understanding this region is vital to ITER operation, fusion reactor design and optimisation
- Complex nonlinear physics: plasma turbulence, neutrals, impurities, material surfaces
- Complex geometry

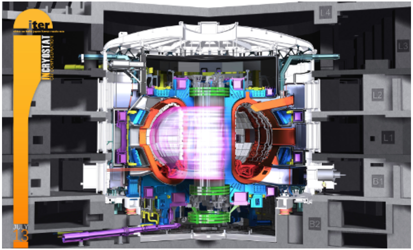


Figure: ITER Organisation

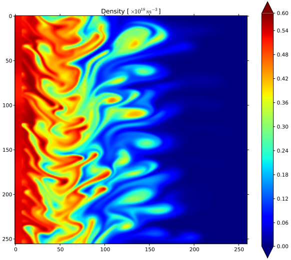


Figure: 2D BOUT++ edge turbulence simulation

BOUT++ overview

What BOUT++ is:

- A toolbox for solving PDEs on parallel computers. Aims to reduce duplication of effort, and allow quick development and testing of new models
- A collection of examples and test cases
- Focussed on flute-reduced plasma models in field-aligned coordinate systems, though has more general capabilities

And is not:

- A single plasma model or simulation
- A general library of numerical methods. Other libraries like PETSc are available for that
- Magic. Appropriate numerical methods depend on the problem, and must be chosen intelligently by the user

BOUT++: A toolbox for plasma simulations

- Collection of useful data types and associated routines. Occupies a middle ground between problem-specific codes and general libraries (e.g. PETSc, Trilinos, Overture, Chombo,...)
- Has its origins in the BOUT code[1][2][3]. Re-written and re-designed (at least once) in C++[4][5]
- Researchers can make use of a (mostly) well tested library of simulation code and input / output tools
- Greatly reduces the time needed to develop a new simulation

---

[1] X.Q. Xu and R.H. Cohen, Contrib. Plasma Phys. 38, 158 (1998)
[2] Xu, Umansky, Dudson, Snyder, CiCP 4, 949-979 (2008)
[3] Umansky, Xu, Dudson et al. Comp. Phys. Comm. 180, 887-903 (2008)
[4] Dudson, Umansky, Xu et al. Comp. Phys. Comm 180, 1467 (2009)
[5] Dudson et al. to appear J. Plasma Phys. (2014).
Preprint: `http://arxiv.org/abs/1405.7905`

BOUT++ key features

- Finite difference initial value code in 3D
- Implicit or explicit time integration methods
- Coordinate system set in metric tensor components
- Handles topology of multiple X-points
- Written in C++, quite modular design
- Open source (LGPL), publicly available

To get a copy:

```
git clone https://github.com/boutproject/BOUT-dev.git
```

Growing BOUT++ community

- Open source project, available on github

```
http://boutproject.github.io
```

- Users in labs and universities in UK, US, China, India, Japan, S.Korea, Denmark,...

A number of fluid and gyrofluid models have been developed using BOUT++ and applied mainly to ELM simulations

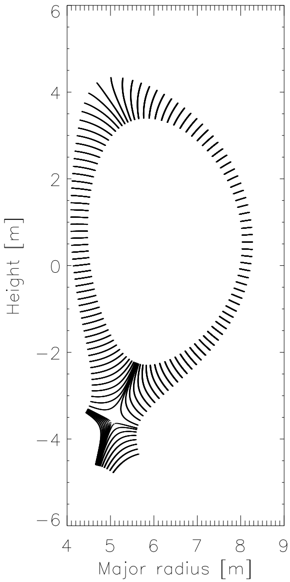
Fluid	Gyrofluid	Physics
3-field $(\omega, P, A_{\parallel})$	1 + 0 $(n_{iG}, n_e, A_{\parallel})$	Peeling-ballooning mode
4-field $(\omega, P, A_{\parallel}, V_{\parallel})$	2 + 0 $(n_{iG}, n_e, A_{\parallel}, V_{\parallel})$	+ acoustic mode
5-field $(\omega, P, A_{\parallel}, T_i, T_e)$		+ Thermal transport no acoustic wave
6-field $(\omega, P, A_{\parallel}, V_{\parallel}, T_i, T_e)$	3 + 1 $(n_{iG}, n_e, A_{\parallel}, V_{\parallel},$ $T_{i\perp}, T_{i\parallel}, T_e)$ Snyder-Hammett model	+ additional drift wave instabilities

BOUT++ geometry

- The BOUT++ coordinate system is field-aligned:

$$
\begin{aligned}
x &= \psi - \psi_0 \qquad y = \theta \\
z &= \phi - \int_{\theta_0}^{\theta} \frac{B_\phi}{RB_\theta} dl_\theta
\end{aligned}
$$

  hence the y unit vector $\hat{e}_y$ is along the magnetic field. This coordinate system has a singularity at the X-point
- A flux-surface aligned mesh is used, similar to that used by e.g. SOLPS
- The domain therefore has a branch-cut, and hole at the X-point itself



Numerical methods - Time integration

- Method of Lines, either fully explicit or fully implicit[1]
- Implicit time integration methods operate Jacobian-free
- Users only need to implement the (nonlinear) function $F(\cdot)$ which operates on state vector $\mathbf{f}$:

$$\frac{\partial \mathbf{f}}{\partial t} = F(\mathbf{f})$$

- Standard explicit methods e.g. RK4, Karniadakis with or without adaptive timestepping
- Implicit methods through the PETSc and SUNDIALS libraries
  $\rightarrow$ **Most simulations use:** adapive order, adaptive timestep BDF method from SUNDIALS
- Optional preconditioning also implemented. Problem specific, usually "physics-based" preconditioner

[1] Some work on split operator, IMEX schemes ongoing

Numerical methods - Spatial operators

A number of operators are available, which users combine to implement models e.g

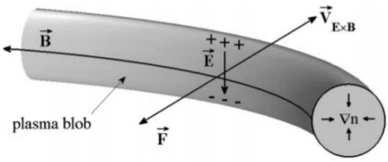
$$\frac{\partial n}{\partial t} = -[\phi, n] + 2\frac{\rho_s}{R_c}\frac{\partial n}{\partial z} + D_n\nabla_\perp^2 n$$

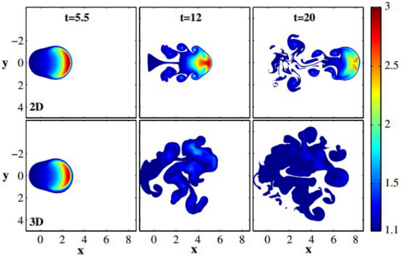
becomes:

```
ddt(n) = - bracket(phi, n, BRACKET_ARAKAWA)
         + 2*DDZ(n)*(rho_s/R_c)
         + D_n*Delp2(n)
```

- The method to be used can be set globally (input file or command-line), per-dimension, or per-operator
- Diffusion-like operators: 2^{nd}-order, 4^{th}-order central difference or 3^{rd}-order CWENO
- Advection operators: Arakawa, CTU, 1^{st}-order and 4^{th}-order upwind, 3^{rd}-order WENO

3D simulations of blobs using BOUT++[2,3]





3D drift waves play an important role in breaking up blobs

$$
\begin{aligned}
\frac{\partial n}{\partial t} + \mathbf{v}_E \cdot \nabla n &= 2c_s\rho_s\left(\mathbf{b}\times\kappa\right)\cdot\nabla n + \nabla_{\|}\frac{J_{\|}}{e} \\
\rho_s^2 n \frac{d}{dt}\nabla_{\perp}^2\phi &= 2c_s\rho_s\left(\mathbf{b}\times\kappa\right)\cdot\nabla n + \nabla_{\|}\frac{J_{\|}}{e} \\
J_{\|} &= \frac{\sigma_{\|}T_e}{en}\left(\nabla_{\|}n - n\nabla_{\|}\phi\right)
\end{aligned}
$$

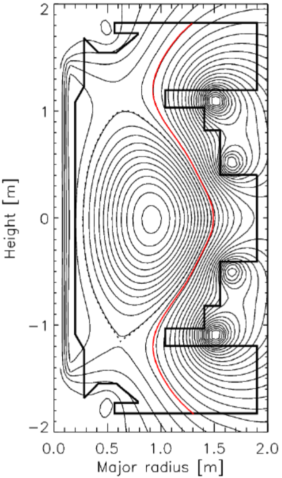
[2] J.R.Angus, M.V.Umansky, S.I.Krashenninikov PRL **108** 215002 (2012)

[3] J.R.Angus, M.V.Umansky, S.I.Krashenninikov Contrib. Plasma Phys. **52** 348-352 (2012)

Application: Flux-tube simulations

Influence of X-point region on blob dynamics can be studied in flux-tube geometry – previous work by N.Walkden on MAST[4]

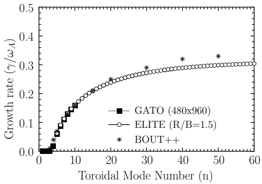
- Shear in X-point region found to decouple midplane and divertor dynamics
- Has been extended to study Super-X configurations
  $\rightarrow$ N.Walkden PSI poster
- Would initially model effect of configuration on **density** transport, not thermal energy
- More sophisticated model includes thermal transport

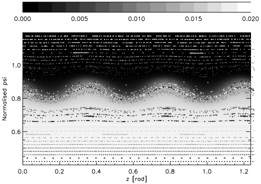


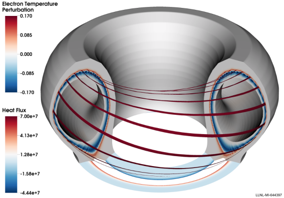
See John's talk

[4] N R Walkden, B D Dudson, G Fishpool PPCF 55 (2013) 105005

- Good agreement with ELITE for linear growth-rates (at least for ballooning modes)
- Linear stability of Snowflake configurations
- Nonlinear multi-mode simulation of ELMs in limiter and X-point geometry
- Importance of small-scale dissipation in determining the size of ELM crash (reconnection).







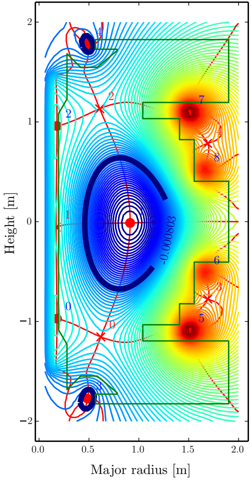
T.Y. Xia *et al.* Nucl. Fusion 53 073009 (2013)
X.Q. Xu *et al.* Physics of Plasmas 20, 056113 (2013)

J.F. Ma *et al.* Nucl. Fusion 54 033011 (2014)

Key developments since 2013 BOUT++ workshop

- Improvements to boundary conditions
- Verification using the Method of Manufactured Solutions
- Implementation of Flux Coordinate Independent (FCI) method
- Improved tokamak geometry mesh generation in IDL and Python
- Coupling to SLEPSc for eigenvalue calculations [Experimental]
- Numerous small improvements and bug fixes

$\sim$ 10,400 additional lines of code



BOUT++ users at the University of York

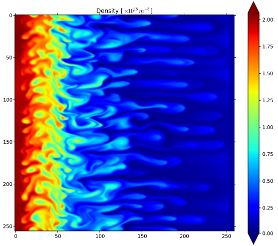
- Ben Dudson [Uni. York staff]
- David Dickinson [EFDA Fellow]
- Samad Mekkaoui [EFDA Fellow, with Jülich]
- Jarrod Leddy [PhD student, with CCFE]
- Brendan Shanahan [PhD student]

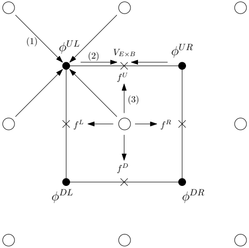
Joint supervision of PhD students based at CCFE:

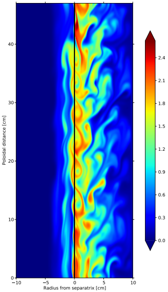
- Nicholas Walkden
- Luke Easy

B.Dudson: Edge turbulence and neutral gas

- 2D and 3D simulations of tokamak edge turbulence, and improving predictions of divertor fluxes
- Interaction of neutral gas with edge plasmas
  - Divertor detachment dynamics
  - Recycling and fuelling of the plasma edge
- Improving numerical methods for edge simulations

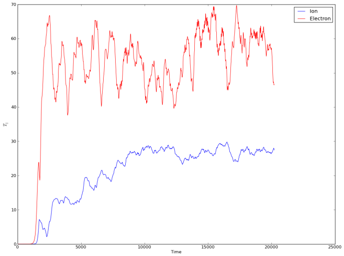






D.Dickinson: Gyro-fluid modelling

- Implementing GEM gyrofluid model [arXiv:0710.4899v3]
    - Electrostatic done
    - Still need to finish $A_{||}$ calculation (may involve code addition to BOUT++)
    - Benchmark in core conditions with GS2 then push outwards

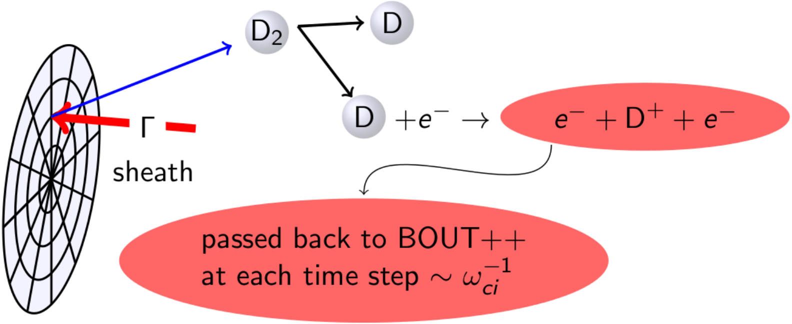


- Adding SLEPSc solver implementation to find eigenvalues
    - Implemented and (seems to) work. Need to find best way to integrate with rest of BOUT++
    - Useful for linear studies, analysing impact of artificial terms, finding time step limits etc.

Samad Mekkaoui: BOUT++ / EIRENE coupling

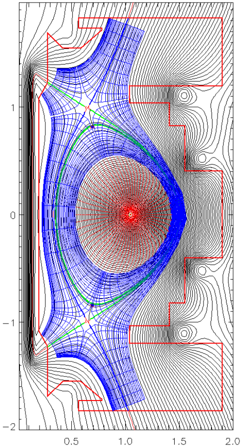
Work ongoing to couple BOUT++ to EIRENE

- Starting in linear geometry (LAPD, PSI-2, Magnum-PSI)
- Technical coupling completed. Both codes run in parallel (MPI), passing data in memory
- Initial turbulence simulations beginning
  $\rightarrow$ Some issues related to initialisation transients



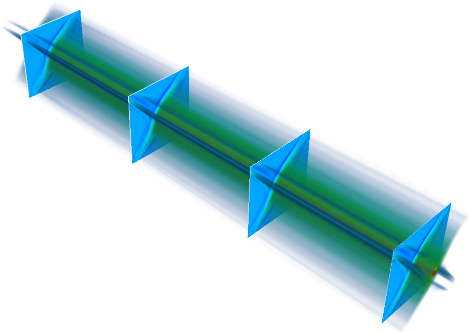
J.Leddy: Drift reduction and core-edge coupling

- Analysis of large eddy simulation techniques for fluid models (Hazeltine and Hasegawa-Wakatani)
- Development of non-orthogonal grid metrics for use in BOUT++ to use simulation grids that match realistic diverter geometries
- Simulations to confirm analytic linear growth rates for drift-wave instabilities in drift-reduced and full velocity fluid models
- Pairing of BOUT++ and CENTORI for 3D, 2-fluid integrated modelling of core and edge



B.Shanahan: Modelling of X-point configurations

- Influence of magnetic X-points on blob propagation and turbulence
- Currently using drift-reduced cold ion models
- Studies in linear geometry for comparison with experiment
- Extensions to toroidal geometry: Verify against Torpex experiments
- Aim is simulation of tokamak X-point regions



Purpose of this workshop

- To discuss key physics and computational issues related to the edge of fusion devices
- To share new developments, and prepare researchers to use and further develop BOUT++
- To promote collaborations within the BOUT++ community and beyond