Trinity Multiscale Transport Code Development for Experimental Comparison

E. G. Highcock[1,2], M. A. Barnes [1,2], G. Colyer[1,2], J. Citrin[3,4], D. Dickinson[2,5], N. R. Mandell[6], L. F. van Wyk[1,2], C. M. Roach[2], A. Jackson[7], J. Hein[8], H. Leggate[9], L. Anton[10], A. A. Schekochihin[1] and W. Dorland[6]

[1] Rudolf Peierls Centre for Theoretical Physics, University of Oxford, United Kingdom
[2] CCFE, Culham Science Centre, Abingdon, Oxon, OX14 3DB, United Kingdom
[3] FOM Institute DIFFER, Trilateral Euregio Cluster, 3430 BE Nieuwegein, The Netherlands
[4] CEA, IRFM, F-13108 Saint Paul Lez Durance, France
[5] York Plasma Institute, Department of Physics, University of York, Heslington, York, YO10 5DD, UK
[6] Department of Physics, University of Maryland, College Park, Maryland, 20742, USA
[7] EPCC, The University of Edinburgh, Kings Buildings, Mayfield Road, Edinburgh, EH9 3JZ, UK
[8] Lunds Universitet, Box 118, 221 00 Lund, Sweden
[9] Dublin City University, Glasnevin, Dublin 9, Ireland.
[10] STFC Daresbury Laboratory, Keckwick Lane, Daresbury, Cheshire, WA4 4AD, UK

e-mail: *edmund.highcock@physics.ox.ac.uk*

Summary

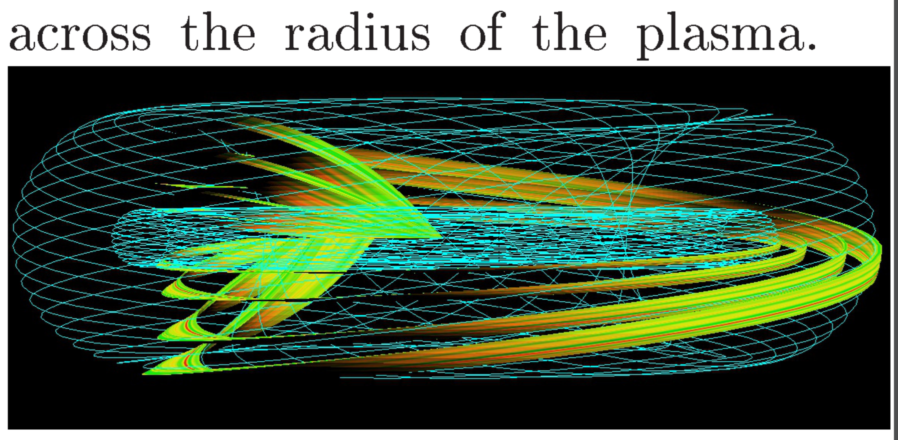
The Trinity multiscale transport code has been extensively upgraded to further its use in experimental comparison. The upgrades to Trinity have extended its capability to work with experimental data, allowed it to evolve the magnetic equilibrium self-consistently (at fixed current) and significantly enhanced the range and performance of its turbulent transport modeling options.

Trinity

Trinity (Barnes et al., 2010) is a multiscale transport code which uses a gyrokinetic code such as Gs2 or Gene to calculate the transport resulting from the turbulence over a short timewindow, and then uses the results to evolve the profiles of temperature, density and flow over a long transport timescale. For example, equation (1) relates the evolution of the density to the time and flux surface average of the turbulent particle transport.

$$\frac{\partial n_s}{\partial t} = -\frac{\partial \psi}{\partial V}\frac{\partial}{\partial \psi}\left[\frac{\partial V}{\partial \psi}\left\langle\left\langle\int d^3\mathbf{v}\left(\mathbf{v}_\chi\cdot\nabla\psi\right)h_s\right\rangle\right\rangle\right] \quad (1)$$

Trinity can either be coupled to a global code, which covers the whole radius, or a local code: the image to the right shows multiple Gs2 flux tubes being used to sample the turbulence across the radius of the plasma.

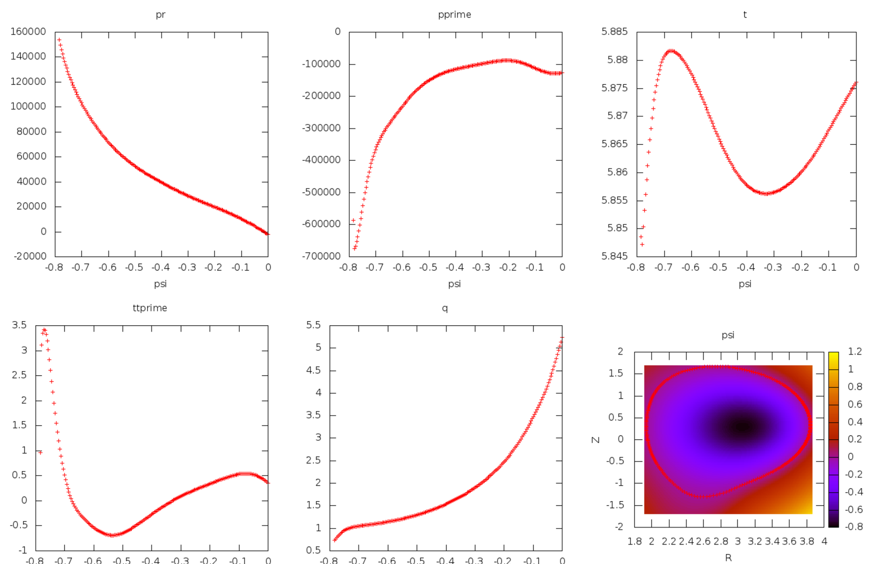


Coupling to Chease

Trinity has been coupled to the Chease equilibrium solver (Lütjens, Bondeson, and Sauter, 1996). Chease is a code which solves the Grad-Shafranov equation (2), which relates the poloidal magnetic field to the toroidal magnetic field and the pressure. The image shows a solution to the Grad-Shafranov equation generated by Chease for use with trinity.

$$\nabla\cdot\frac{1}{R^2}\nabla\Psi = \frac{j_\phi}{R} = -p'(\Psi) - \frac{1}{R^2}ff'(\Psi) \quad (2)$$

A Trinity simulation can simply take the geometry from a Chease file and find a steady state solution, or it can rerun Chease at every timestep to create a magnetic equilibrium consistent with the new pressure profile (at constant radial profile of the flux surface averaged current).

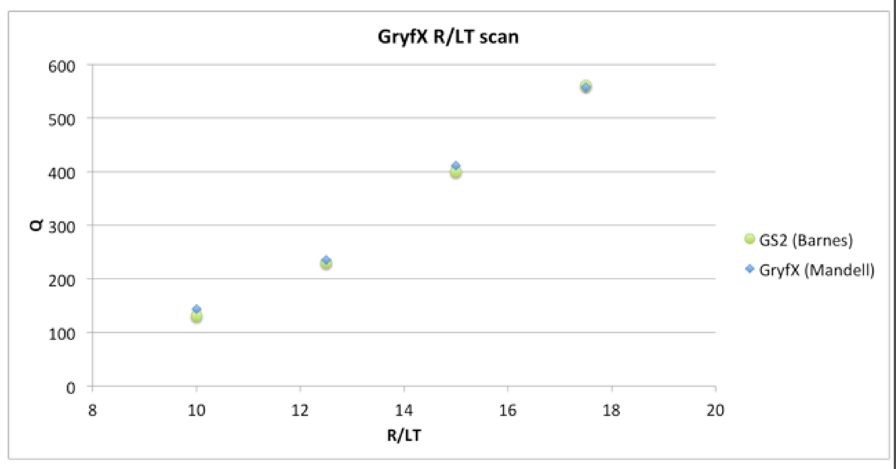


Input from Cronos

The Cronos code (Artaud et al., 2010) can be used to provide highly accurate reconstructions of experimental tokamak discharges. Work has been carried out to upgrade Cronos to work on modern 64-bit systems. Following this, new export options have been created for Cronos to allow Trinity to use equilibrium reconstructions from Cronos. Trinity reads in geometric data from Cronos and uses the information to generate a Miller equilibrium for the turbulence codes. This process has been tested extensively by comparing the results where the geometry from Cronos is passed directly to Gs2.

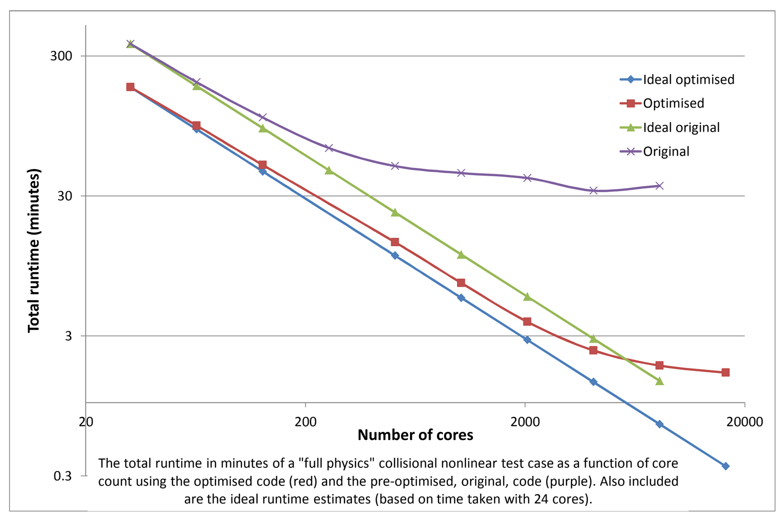
Coupling to GryfX

When using a gyrokinetic code to calculate the properties of the turbulence, Trinity uses a very large amount of computer time, owing to the expense of the gyrokinetic simulations. To partially solve this problem, Trinity has been recently coupled to a code, GryfX, which solves the gyrofluid equations (Dorland, 1993; Beer, 1995) on a graphics processing unit (GPU). This code, which can also be used to calculate the properties of turbulent transport, runs many times faster than a gyrokinetic code, and can take advantage of cheap and plentiful GPU processing power. It is anticipated that GryfX will be used to run searches of parameter space, before using gyrokinetic simulations to confirm results.



Performance upgrades to Gs2

An ongoing program has made significant progress in improving the performance of Gs2, one of the main codes used to calculate turbulent transport within Trinity. This progress, which has been recognised by a 2014 HPC Innovation Excellence Award, has included changes in memory layout and partial load imbalancing (Jackson, Hein, and Roach, 2014), as well as the systematic targeting of bottlenecks and rewrites of problematic sections of code such as the solution of the field equations. This has produced large increases in performance (shown in the image, courtesy of D. Dickinson) which will benefit Trinity simulations strongly.



References

Artaud, J.F. et al. (Apr. 2010). *Nuclear Fusion.* "The CRONOS suite of codes for integrated tokamak modelling".

Barnes, M. et al. (2010). *Physics of Plasmas.* "Direct multiscale coupling of a transport code to gyrokinetic turbulence codes".

Beer, M.A. (1995). "Gyrofluid models of turbulent transport in tokamaks".

Dorland, W.D. (1993). "Gyrofluid models of plasma turbulence".

Jackson, Adrian, Joachim Hein, and Colin Roach (2014). *IEEE Transactions on Parallel and Distributed Systems.* "Optimising Performance Through Unbalanced Decompositions".

Lütjens, H., A. Bondeson, and O. Sauter (1996). *Computer physics communications.* "The CHEASE code for toroidal MHD equilibria".

Acknowledgements

This work was supported by STFC and the Culham Centre for Fusion Energy. Computing time was provided by IFERC grants MULTEI and GKDELB, The Hartree Centre, and EPSRC grants EP/ H002081/1 and EP/L000237/1. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 633053 and from the RCUK Energy Programme [grant number EP/I501045]. The views and opinions expressed herein do not necessarily reflect those of the European Commission.