First results from core-edge parallel composition in the FACETS project

J R Cary[1], J Candy[2], R H Cohen[3], S Krasheninnikov[4], D C McCune[5], D J Estep[6], J Larson[7], A D Malony[8], A. Pankin[9], P H Worley[10], J A Carlsson[1], A H Hakim[1], P Hamill[1], S Kruger[1], M Miah, S Muzsala[1], A Pletzer[1], S Shasharina[1], D Wade-Stein[1], N Wang[1], S. Balay[7], L McInnes[7], H Zhang[7], T Casper[3], L Diachin[3], T Epperly[3], T D Rognlien[3], M R Fahey[10], J Cobb[10], A Morris[8], S Shende[8], G W Hammett[5], K Indireshkumar[5], D Stotler[5], A Yu Pigarov[4]

[1]Tech-X Corporation, Boulder, CO 80303, USA
[2]General Atomics, San Diego, CA 92186, USA
[3]Lawrence Livermore National Laboratory, Livermore, CA 94550, USA
[4]University of California at San Diego, San Diego, CA 92093, USA
[5]Princeton Plasma Physics Laboratory, Princeton, NJ 08543, USA
[6]Colorado State University, Fort Collins, CO 80523, USA
[7]Argonne National Laboratory, Argonne, IL 60439, USA
[8]ParaTools, Inc., Eugene, OR 97405, USA
[9]Lehigh University, Bethlehem, PA 18015, USA
[10]Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

E-mail: cary@txcorp.com

Abstract. FACETS (Framework Application for Core-Edge Transport Simulations), now in its second year, has achieved its first coupled core-edge transport simulations. In the process, a number of accompanying accomplishments were achieved. These include a new parallel core component, a new wall component, improvements in edge and source components, and the framework for coupling all of this together. These accomplishments were a result of an interdisciplinary collaboration among computational physics, computer scientists, and applied mathematicians on the team.

1. Introduction

The FACETS (Framework Application for Core-Edge Transport Simulations) project [1] has the goal of providing whole-tokamak modeling through coupling separate components for each of the core region, edge region, and wall. A coupling approach is used because (1) there exist computational models for each region, (2) the regions are covered by different approximations, and (3) direct simulation resolving the gyrokinetics time scales in all regions are as too computationally intensive. That is, even when the code is parallelized, computing 1 ms of experimental time using gyrokinetics requires 1 hour of computational time. Thus, computing the full 1000-second discharge would require 10^6 hours.

FACETS has organized multidisciplinary teams for each of the components and for the framework. The computational fusion scientists provide the model and basic component, but in working with applied mathematicians and computers scientists, the implementations are improved. Computer scientists are also involved in interface definition and interlanguage implementation.

© 2008 IOP Publishing Ltd

2. Core solver including performance analysis

Tokamak transport is dominated by instability-driven turbulence at extremely small spatial scales. The resultant turbulent transport is highly nonlinear and increases rapidly beyond the instability threshold in temperature gradient. A commonly used model of this turbulent transport uses fitted quasilinear theory to model this transport and has shown success in modeling tokamak plasmas [2]. The difficulty of using this model is the length of computation in calculating the fluxes, which is a separate computation at each radial cell, and in advancing the integrated system with these nonlinear, radially dependent fluxes.

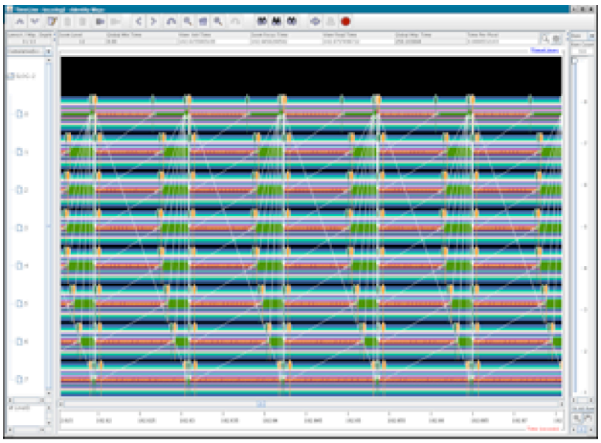

Figure 1. Using the TAU performance system [3] for performance analysis has proven critical in achieving scalable performance.. Shown here is an example of how unbalanced flux calculations (purple) lead to wasted CPU cycles (green) as the time advance requires all of the flux calculations to complete. This lead to a change in the code that allows for more balanced calculations.

To solve these problems, a new core solver was written from scratch using the FACETS infrastructure. As discussed below, the new solver has been verified against established codes, but the new code has allowed us to parallelize the flux calculations and achieve a factor of 30 speedup on 128 processors with a 128-cell grid. The time advance uses a Newton solve with a multilevel method to enlarge the radius of convergence. The overall performance is sensitive to the load balancing of the flux calculations, and performance analysis has proven critical to identifying these problems, as shown in figure 1. As part of an embedded SAP, these flux calculations will be replaced by even more intensive first-principles (gyrokinetic) calculations. The new code has been written to allow us to fully exploit leadership computers as the individual calculations increase in computational cost. By improving the parallelism and algorithms, we hope to perform full 1000-second ITER simulations in less than an hour.

3. Componentization and parallelization of UEDGE

The model for the region from the edge pedestal, through the magnetic separatrix, and to material walls is the existing 2D UEDGE transport code that includes electrons, multi-species ions, and neutrals. UEDGE is the major US-developed edge transport code for modeling the tokamak boundary plasma. Figure 1 shows a typical mesh and electron temperature solution. The complexity of the amount of physics that UEDGE encapsulates has meant that users prefer a python interface for control of UEDGE. While this has been a benefit for users, it has introduced additional interlanguage problems with the C++ framework, described below.

UEDGE has also previously used a Newton-Krylov solver and preconditioning technique, where the latter does not provide any domain-overlap strategy for the parallel UEDGE option that has been revived during this project. Being 2D, edge transport can be a performance-limiting component of the coupled system. To remedy this deficiency, we developed an interface to the nonlinear solvers (SNES) in PETSc. As anticipated, in the serial mode, SNES performance is close to the previous solver. PETSc has allowed the easy exploration of various parallel preconditioning strategies, which show improved convergence efficiency.

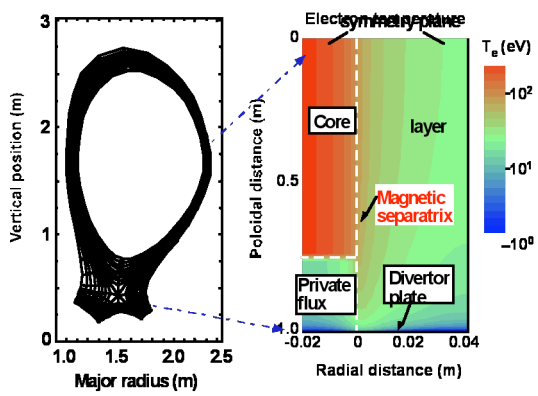

Figure 2. Typical UEDGE mesh and electron temperature showing 2D variation in the outer edge, but 1D variation at the left boundary where core model is coupled.

4. Core sources

Fusion experiments plan to reach ignition temperatures by using external sources: both neutral beam injection and radiofrequency heating. Modeling of these external sources is done by using parallel codes, and can consume considerable resources by itself. The recently parallelized PPPL Monte Carlo package, NUBEAM [2, 3] will be used for providing the core sources from neutral beam injection (NBI) and fusion reactions. The output of NUBEAM includes particle sources (both charged and neutral particles), and momentum and energy source rates along with a wide array of other output including the slowing down fast ion distribution functions. As part of the FACETS project, NUBEAM is undergoing performance analysis and scaling testing. Since the only communications costs in NUBEAM involve the broadcast of initial data and the MPI reductions at the end of deposition and orbiting, good scaling is expected. Recent tests demonstrate excellent scaling up to 128 processors with 1 million particles. Efforts are under way to test this scaling further.

As part of the Simulation of Wave Interactions with MHD (SWIM) SciDAC, common interfaces for all sources were defined, and an API using Fortran was defined. As part of the FACETS project, we have extended this API to include C++ interfaces. The solution of the problem of how to efficiently construct this interface led to a new publication on Fortran90/C++ interoperability. With this new interface, we will be able to leverage the work of the SWIM project and incorporate RF and particle sources into our parallel core solver. Ahead is yet to work out the interfaces for sinks, such as radiation.

5. Framework and coupling

We have used the framework to obtain initial results for the core-edge coupled system. In these simulations the core component is native to FACETS; that is, it is written by using infrastructure provided by the framework itself. The edge component is a non-native component, obtained by wrapping UEDGE using Babel [6]. The framework now provides an explicit coupler component which calls the individual components to advance their internal state (solution) by a given time step. The coupler then exchanges the core-edge boundary data and loops to move the solution forward in time. Current efforts are directed toward the coupled simulation results and the development of an implicit coupler component

6. The wall component

Plasma-surface interactions are complex due to the complex chemistry and plasma physics that occurs on a short timescale. To model these complex interactions, we have developed the Wall and Plasma-Surface Interactions (WallPSI) module, which is the 1-D multiscale, multispecies code for particle and heat transport inside plasma-facing components [7]. The code incorporates new approaches in continuum modeling of hydrogen species in wall materials. The code is being validated against the vast experimental data on hydrogen retention, permeation, and erosion rates for major fusion related materials.

Preliminary results of self-consistent plasma-neutrals-wall modeling were reported for the first time in [7] using WallPSI/Edge1D and showed examples of strong plasma-wall coupling, nonlinear variation of wall hydrogen inventory and recycling coefficient with respect to incident plasma flux, and the featured plasma instabilities. The modeled transitional effects include sawteeth-like oscillations in edge plasma due to switching from wall pumping to gas release occurred in the case without external gas puff/pump feedback, where 15–20% variation in recycling coefficient corresponds to transitions between cold deeply-detached and hot sheath-limited edge plasmas due to thermal plasma instability [8].

Ongoing work on WallPSI integration in FACETS includes development of the infrastructure and plasma-wall interface, implementation of robust nonlinear solvers, and massive parallelization via the FACETS framework based on assigning one CPU to each wall segment of the modeling domain. The coupling of WallPSI to UEDGE (not completed) will be through assigning one instance of WallPSI to each wall facing UEDGE.

6. Steady-state gyrokinetic transport

FACETS has an embedded Scientific Application Partnership (SAP) whose goal to develop a strategy to employ local and global gyrokinetic turbulence simulations in steady state transport calculations. A prototype transport driver has been developed that can run multiple instances of GYRO, a fundamental physics gyrokinetic code, or simpler transport models. This state-of-the-art code computes self-consistent transport using a Newton iteration scheme. Early results show that using this iteration scheme a fairly crude steady state can be obtained in just a few iterations. Ongoing work on the physics of this iteration scheme includes investigating the Newton step-size as well as the GYRO time-averaging length.

Our prototype simulation was a 12-hour simulation run on the Cray XT4 at Oak Ridge National Laboratory using 1284 cores. This simulation used 25 radial grid points over the minor radius: 4 instances of a simpler model were run over the range 0.0375 to 0.15 while 20 instances of GYRO each using 64 processors were run over 0.1875 to 0.9. This could easily have been run with more grid points and with each GYRO instance using more processes, thus the scaling to very high numbers of cores is straightforward. Current efforts are leveraging this embedded SAP to be made available to the new core solver so that this parallelism can be efficiently exploited in a time-dependent simulation.

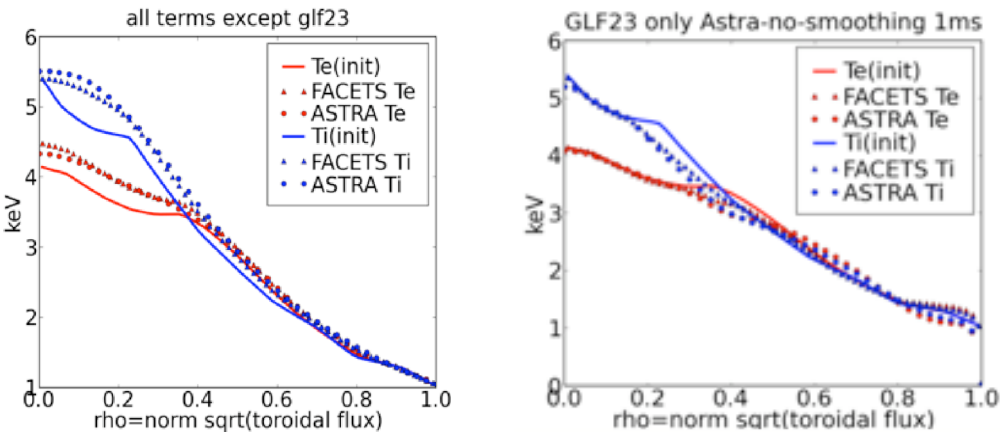

Figure 3. Comparison of the FACETS core solver with the ASTRA code evolved an experimental discharge for 1 microsecond with different terms for energy transport used. On the left, no turbulent transport was considered. The codes agree well except near the core, where the ASTRA code does not conserve energy. The case with turbulent transport also agrees well except for problems near rho of 0.4, believed to be due to time-advance differences.

7. Verification of the core solver

Because the new core solver was written from scratch, verifying the code is critical to ensure its accuracy. To verify the code, the code was compared to the ASTRA transport code, a standard transport code used in Europe. The code was compared both with and without the turbulence transport model described above to isolate problems. The disagreement for the case without the turbulent transport was traced to a lack of energy conservation in the ASTRA code. Disagreements also exist for the case with the turbulent transport, and the differences are believed to be due to different time advance schemes (ASTRA uses an explicit scheme). Efforts are under way to refine these verification efforts by using a steady-state code for comparison.

8. Concluding remarks

FACETS has achieved core-edge coupled simulations in its first year through development of a multicomponent framework that allows separate components to run on different processors. In the process FACETS developed a new, parallel core solver, restored parallelism to the edge component, and implemented a wall model. Applied mathematics and computer science have been an integral part of this project through algorithms, performance analysis, and interlanguage operability.

In its initial phase, FACETS has proceeded with the philosophy of maximally reusing code that is available, while writing new components and infrastructure where warranted. Furthermore, FACETS has worked to provide developed modules to other teams. The sister FMCFM project has defined component interfaces for turbulent transport models, and these are being shared, in particular, with the SWIM project, which has another approach to component integration.

9. Future directions

FACETS has only just started, and so many tasks remain. One of these is to make the core-edge coupling application a robust, user application by improving the initial setup of the edge component. Others include the move to implicit coupling, incorporation of the wall component, bringing in core sources, allowing for dynamic magnetic equilibrium, validation studies, and releasing version 1.0. In addition, we will incorporate the work of the SSGKT SAP program for more fundamental flux calculations.

As well, we will have to continue improvements of our code base. We expect to make further refinements to our interfaces. Non-native components bring a challenge in that they were typically written without the needs of componentization in mind; for example, they often make heavy use of global variables and so are not thread safe. This latter point makes it impossible to make use of mixed parallelism. Another problem is the used of shared objects, as is needed for python-wrapped codes. Shared libraries are not available on the reduced operating systems on the compute nodes of leadership class supercomputers. These challenges require a continual evaluation of component requirements and project needs as we progress. Coupling codes with different mathematical models brings new challenges in ensuring the reliability of the results. We will extend the a posteriori analysis of finite volume methods to time dependent problems using various time integration schemes, including a detailed study of stability issues. Then, we will analyze the effect of coupling through a common boundary, including the interaction with the time integration. We also will analyze the effects of using nonuniform cells on the accuracy of the finite volume method.

Of course, the primary challenge facing FACETS, with its parallel component (MCMD) decomposition model, is effectively distributing the components over the expected 10^5 nodes of a high-performance computer.

Acknowledgments

Work supported by the U.S. Department of Energy (DOE) Office of Science under grants and contracts including DE-FC02-07ER54907 at Tech-X, DE-AC02-06CH11357 at ANL, DE-FC02-07ER54909 at CSU, DE-AC52-07NA27344 at LLNL, DE-AC05-00OR22725 at ORNL, DE-FC02-07ER54910 at ParaTools, DE-AC02-76CH03073 at PPPL, DE-FC0207-ER54908 and DE-FG0204-ER54739 at UCSD. This work used the resources of the National Energy Research Scientific Computing Center, which is supported by the DOE Office of Science under Contract No. DE-AC02-05CH11231, and of the National Center for Computational Sciences at ORNL, which is supported by the DOE Office of Science under Contract No. DE-AC05-00OR22725.

References

[1] https://www.facetsproject.org/facets
[2] Kinsey J E, Staebler G and Waltz R E 2003 *Fus. Sci. Tech.* **44** 763
[3] Shende S and Malony A 2006 The TAU parallel performance system *Int. J. High Perf. Comp. Apps.* **20(2)** 287–331, http://tau.uoregon.edu.
[4] Goldston R J, *et al.* 1981 *J. Comp. Phys.* **43** 61
[5] Pankin A, *et al.* 2004 *Comp. Phys. Comm.* **159** 157
[6] https://computation.llnl.gov/casc/components/babel.html
[7] Pigarov A Yu, *et al.* 2008 Coupled plasma-wall modeling *Proc. 18th Int. Conf. on Plasma Surface Interactions* (Spain)
[8] Krasheninnikov S I *et al.* 2006 *Phys. Plasmas* **13** 094502