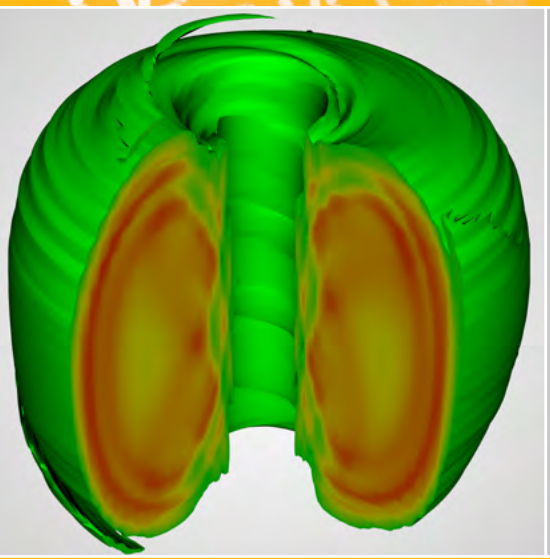
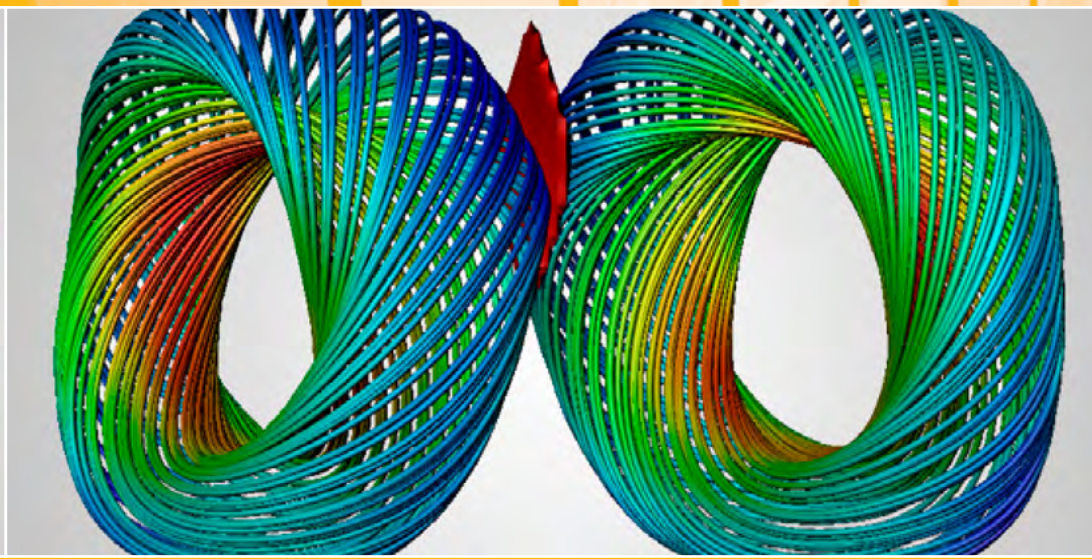
Large Scale Computing and Storage Requirements for Fusion Energy Sciences: Target 2017

Report of the NERSC Requirements Review
Conducted March 19–20, 2013





NERSC

BERKELEY LAB

U.S. DEPARTMENT OF ENERGY
Office of Science

Draft

DISCLAIMER

This report was prepared as an account of a workshop sponsored by the U.S. Department of Energy. Neither the United States Government nor any agency thereof, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Copyrights to portions of this report (including graphics) are reserved by original copyright holders or their assignees, and are used by the Government's license and by permission. Requests to use any images must be made to the provider identified in the image credits.

Ernest Orlando Lawrence Berkeley National Laboratory is an equal opportunity employer.

Ernest Orlando Lawrence Berkeley National Laboratory

University of California

Berkeley, California 94720 USA

NERSC is funded by the United States Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) program. David Goodwin is the NERSC Program Manager and John Mandrekas serves as the Fusion Energy Sciences (FES) allocation manager for NERSC.

NERSC is located at the Lawrence Berkeley National Laboratory, which is operated by the University of California for the U.S. Department of Energy under contract DE-AC02-05CH11231.

This work was supported by the Directors of the Office of Science, Office of Fusion Energy Sciences, and the Office of Advanced Scientific Computing Research, Facilities Division.

Draft

Large Scale Computing and Storage Requirements for Fusion Energy Sciences: Target 2017

Report of the HPC Requirements Review

Conducted March 19-20, 2013

Rockville, MD

DOE Office of Science

Office of Fusion Energy Sciences (FES)
Office of Advanced Scientific Computing Research (ASCR)

National Energy Research Scientific Computing Center (NERSC)

Editors

Richard A. Gerber, NERSC

Harvey J. Wasserman, NERSC

Draft

Large Scale Computing and Storage Requirements for Fusion Energy Sciences: Target 2017

Table of Contents

1 Executive Summary 5

2 DOE Fusion Energy Sciences Mission 6

3 About NERSC 8

4 Meeting Background and Structure 9

5 Meeting Demographics 10
- 5.1 Participants 10
- 5.2 NERSC Projects Represented by Case Studies 13

6 Findings 15
- 6.1 Summary of Requirements 15
- 6.2 Additional Observations 16
- 6.3 Requirements Summary 16

7 FES and NERSC Trends 19

8 Magnetic Fusion Energy Science (MFES) Case Studies 20
- 8.1 Transport Calculation and Profile Prediction 20
- 8.2 Energetic Particles and the SciDAC Gyrokinetic Simulation of Energetic Particle Turbulence and Transport Project 25
- 8.3 Centre for Simulation of Wave - Plasma Interaction (CSWPI) 29
- 8.4 Simulation of Plasma Edge Physics 38
- 8.5 Study of the Internal Dynamics of ITER 47
- 8.6 Small Scale Plasma Experiments: Simulations and Model Validation 54

9 General Plasma Science Case Studies 60
- 9.1 Petascale Kinetic Simulations in Laboratory and Space Plasmas 60

10 High Energy Density Laboratory Plasmas (HEDLP) & Inertial Fusion Energy Sciences (IFES) Case Studies 66
- 10.1 Heavy Ion Beams and Interactions with Plasmas and Targets (HEDLP and IFE) 66
- 10.2 Laser Plasma Interactions 76

11 Materials and Plasma-Surface Interactions 80
- 11.1 Plasma Materials Interactions 80

12 Integrated Modeling 86
- 12.1 Integrated Whole-Device Modeling Studies 86

Appendix A. Attendee Biographies 92

Appendix B. Workshop Agenda 98

Appendix C. Abbreviations and Acronyms 100

Appendix D. About the Cover 101

Draft

1 Executive Summary

The National Energy Research Scientific Computing Center (NERSC) is the primary computing center for the DOE Office of Science, serving approximately 4,500 users working on some 650 projects that involve nearly 600 codes in a wide variety of scientific disciplines. In addition to large-scale computing and storage resources NERSC provides support and expertise that help scientists make efficient use of its systems.

In March 2013, NERSC, DOE's Office of Advanced Scientific Computing Research (ASCR) and DOE's Office of Fusion Energy Sciences (FES) held a review to characterize High Performance Computing (HPC) and storage requirements for FES research through 2017. This review is the ninth in a series that began in 2009 and it is the second for FES. The report from the earlier (2010) NERSC FES review is available at http://www.nersc.gov/science/hpc-requirements-reviews/target-2014/.

The latest review revealed several key requirements, in addition to achieving its goal of characterizing FES computing and storage needs. High-level findings are:

1. **To meet Fusion Sciences research objectives, FES researchers need computing and data storage resources in excess of those predicted by historical trends.**
2. **FES codes will require updated mathematical and I/O libraries that will run efficiently on next generation architectures**.
3. **FES scientists need support for both ensemble runs and large-scale jobs.**
4. **Teams need effective tools for managing workflows, performing data analysis, and profiling and developing codes.**
5. **Code teams need help porting applications to run on next-generation architectures.**

This report expands upon these key points and adds others. The results are based upon representative samples, called "case studies," of the needs of science teams within FES. The case study topics and review attendees were selected by the NERSC meeting coordinators and FES program managers to represent the FES production computing workload. Prepared by the FES workshop participants, the case studies contain a summary of science goals, methods of solution, current and future computing requirements, and special software and support needs. Also included are strategies for computing in the highly parallel "many-core" environment that is expected to dominate HPC architectures over the next few years.

2 DOE Fusion Energy Sciences Mission

The mission of the Fusion Energy Sciences (FES) program is to expand the fundamental understanding of matter at very high temperatures and densities and to build the scientific foundation needed to develop a fusion energy source. This is accomplished by studying plasma and its interaction with its surroundings across wide ranges of temperature and density, developing advanced diagnostics to make detailed measurements of its properties and dynamics, and creating theoretical and computational models to resolve the essential physics principles.

FES has four strategic goals:

- Advance the fundamental science of magnetically confined plasmas to develop the predictive capability needed for a sustainable fusion energy source;
- Support the development of the scientific understanding required to design and deploy the materials needed to support a burning plasma environment;
- Pursue scientific opportunities and grand challenges in high energy density plasma science to explore the feasibility of the inertial confinement approach as a fusion energy source, to better understand our universe, and to enhance national security and economic competitiveness, and;
- Increase the fundamental understanding of basic plasma science, including both burning plasma and low temperature plasma science and engineering, to enhance economic competiveness and to create opportunities for a broader range of science-based applications.

From the days of the Controlled Thermonuclear Research (CTR) and the National Magnetic Fusion Energy Computer Center (MFECC) in the mid-1970s—the predecessors of NERSC—high performance computing and NERSC have played a significant role in fusion energy research. Advanced simulations are critical for advancing the FES mission and achieving its goals, especially the development of a predictive capability needed for a sustainable fusion energy source. NERSC resources—distributed via the annual Energy Research Computing Allocations Process (ERCAP) process and supplemented by the ASCR Leadership Computing Challenge (ALCC) program—provide a reliable and predictable resource for meeting critical FES mission needs.

Approximately 75% of the FES NERSC allocation is used to address the needs of the Magnetic Fusion Energy (MFE) Theory and SciDAC programs, which are strongly focused on ITER and burning plasmas. These programs address challenges in macroscopic stability, confinement and transport in the core and edge regions, the interaction of radiofrequency waves with plasmas, and energetic particle effects. A substantial fraction of these resources supports the validation mission of fusion research at the FES major facilities. The rest of the resources are distributed among high energy density plasma science, materials science (a growth area following the addition of a materials science project in the FES SciDAC portfolio), general plasma science, and small-scale experimental plasma research.

In the coming years, the increase in the fidelity and level of integration of fusion simulations needed to resolve multiphysics and multiscale problems in ITER-grade plasmas (including plasma-surface interaction effects), enabled by continuing advances in high performance computing hardware and associated progress in computational algorithms, will demand a significantly higher level of NERSC and other SC high performance computing resources.

Draft

3 About NERSC

The National Energy Research Scientific Computing (NERSC) Center, which is supported by the U.S. Department of Energy's Office of Advanced Scientific Computing Research (ASCR), serves more than 4,500 scientists working on over 650 projects of national importance. Operated by Lawrence Berkeley National Laboratory (LBNL), NERSC is the primary high-performance computing facility for scientists in all research programs supported by the Department of Energy's Office of Science. These scientists, working remotely from DOE national laboratories; universities; other federal agencies; and industry, use NERSC resources and services to further the research mission of the Office of Science (SC). While focused on DOE's missions and scientific goals, research conducted at NERSC spans a range of scientific disciplines, including physics, materials science, energy research, climate change, and the life sciences. This large and diverse user community runs hundreds of different application codes. Results obtained using NERSC facilities are cited in about 1,500 peer reviewed scientific papers per year. NERSC activities and scientific results are also described in the center's annual reports, newsletter articles, technical reports, and extensive online documentation. In addition to providing computational support for projects funded by the Office of Science program offices (ASCR, BER, BES, FES, HEP and NP), NERSC directly supports the Scientific Discovery through Advanced Computing (SciDAC[1]) and ASCR Leadership Computing Challenge[2] Programs, as well as several international collaborations in which DOE is engaged. In short, NERSC supports the computational needs of the entire spectrum of DOE open science research.

The DOE Office of Science supports three major High Performance Computing Centers: NERSC and the Leadership Computing Facilities at Oak Ridge and Argonne National Laboratories. NERSC has the unique role of being solely responsible for providing HPC resources to all open scientific research areas sponsored by the Office of Science.

This report illustrates NERSC alignment with, and responsiveness to, DOE program office needs; in this case, the needs of the Office of Fusion Energy Sciences. The large number of projects supported by NERSC, the diversity of application codes, and its role as an incubator for scalable application codes present unique challenges to the center. However, as demonstrated its users' scientific productivity, the combination of effectively managed resources, and excellent user support services the NERSC Center continues its 40-year history as a world leader in advancing computational science across a wide range of disciplines.

For more information about NERSC visit the web site at http://www.nersc.gov.

[1] http://www.scidac.gov

[2] http://science.energy.gov/~/media/ascr/pdf/incite/docs/Allocation_process.pdf

4 Meeting Background and Structure

In support of its mission to provide world-class HPC systems and services for DOE Office of Science research NERSC regularly gathers user requirements. In addition to requirements reviews NERSC collects information through the Energy Research Computing Allocations Process (ERCAP), workload analyses, an annual user survey, and discussions with DOE program managers and scientists who use the facility.

In March 2013, ASCR (which manages NERSC), FES, and NERSC held a review to gather HPC requirements for current and future science programs supported by FES. This report is the result.

This document presents a number of findings, based upon a representative sample of projects conducting research supported by FES. The case studies were chosen by the DOE Program Office Managers and NERSC staff to provide broad coverage in both established and incipient FES research areas. Most of the domain scientists at the review were associated with an existing NERSC project, or "repository" (abbreviated later in this document as "repo").

Each case study contains a description of current and future science, a brief description of computational methods used, and a description of current and future computing needs. Since supercomputer architectures are trending toward systems with chip multiprocessors containing hundreds or thousands of cores per socket and millions of cores per system, participants were asked to describe their strategy for computing in such a highly parallel, "many-core" environment.

Requirements presented in this document will serve as input to the NERSC planning process for systems and services, and will help ensure that NERSC continues to provide world-class resources for scientific discovery to scientists and their collaborators in support of the DOE Office of Science, Office of Fusion Energy Sciences.

NERSC and ASCR have been conducting requirements workshops for each of the six DOE Office of Sciences offices that allocate time at NERSC (ASCR, BER, BES, FES, HEP, and NP). A first round of meetings was conducted between May 2009 and May 2011 for requirements with a target of 2014; this included an August, 2010 FES review. A second round of meetings, of which this is the third, will target needs for 2017. Reports from all previous NERSC requirements reviews are available on the NERSC web site.

Specific findings from the review follow.

5 Meeting Demographics

5.1 Participants

5.1.1 Organizers

Name	Institution	Area of Interest
Sudip Dosanjh	NERSC	NERSC Director
Richard Gerber	NERSC	Meeting Organizer
Dave Goodwin	DOE / ASCR	NERSC Program Manager
John Mandrekas	DOE / FES	FES Program Manager
Harvey Wasserman	NERSC	Meeting Organizer

5.1.2 Domain Scientists

Name	Institution	Area of Interest	NERSC Repo(s)
Paul Bonoli	MIT	MFES; RF interactions with plasmas	m77
Jeff Candy	General Atomics	MFES; gyrokinetic simulations with continuum (Eulerian) codes; Verification & Validation; Developer of GYRO	m681, m1574, gc3, mp94, m888, mp2
Stephane Ethier	PPPL	MFES; gyrokinetic simulations with PIC codes; code performance issues; co-developer of the GTS code	mp19, m499, m912, gc3
Alex Friedman	LLNL and LBNL	IFES / HEDLP / Heavy Ion Fusion	mp42
Kai Germaschewski	University of New Hampshire	GPS; simulations of magnetic reconnection and turbulence in laboratory and space plasmas	m148
Stephen Jardin	PPPL	MFES; MHD-fluid simulations; developer of the M3D family of codes	m876, mp288, m912, m64, mp21, m1799
Homa Karimabadi	U.C. San Diego	Kinetic simulations of Laboratory and space plasmas; data issues	m1303

Draft

Alice Koniges	NERSC	IFES: Heavy Ion Fusion and Ion Accelerator Experiment Modeling;	mp2, mp42
Arnold Kritz	Lehigh University	Integrated modeling	m649 m876 m1043
Zhihong Lin	U.C. Irvine	MFES; gyrokinetic simulations with PIC codes; developer of GTC code	m808, m499, m92
Vyacheslav Lukin	Naval Research Laboratory	MFES / GPS; development and validation of fluid-based numerical models for simulations of exploratory confinement & basic laboratory plasma research; developer of the HiFi multi-fluid modeling framework	m1255, m489
Alexei Pankin	Tech-X Corporation	ELM dynamics, integrated modeling	mp21, m499, m649, m681, m1043
Scott Parker	U. Colorado	core turbulence and transport; energetic particles; developer of GEM PIC	mp118, gc3
Chuang Ren	University of Rochester	IFES / HEDLP; direct-drive ICF and advanced ignition concepts	m792, mp113, m412
Carl Sovinec	University of Wisconsin	MFES; MHD-fluid simulations; co-developer of the NIMROD two-fluid extended MHD code	mp2, mp200, mp21
Linda Sugiyama	MIT	MFES; MHD-fluid simulations	m224, m499, mp19, mp288
Frank Tsung	UCLA	IFES: Laser-plasma interactions;	mp113, m1157
Brian Wirth	University of Tennessee, and ORNL	Materials; plasma-surface interactions	m1709, m1200, m916
Xueqiao Xu	LLNL	Plasma turbulence	mp2, gc3

5.1.3 Observers

Name	Institution	Area of Interest
Sean Finnegan	DOE / FES	Program Manager, HEDLP, GPS, & Theory
Wayne Joubert	Oak Ridge National Laboratory	Scientific Computing Group Member
Ravinder Kapoor	DOE / ASCR	OLCF Program Manager
Carolyn Lauzon	DOE / ASCR	AAAS S&T Technology Fellow

Draft

Randall Laviolette	DOE / ASCR	SciDAC Application Partnerships
James W. Van Dam	DOE / FES	Director, Research Division
Timothy Williams	ANL / OLCF	Catalyst, laser-plasma interactions

Draft

5.2 NERSC Projects Represented by Case Studies

NERSC projects represented at the review are listed in the table below, along with computing and storage resources each used in 2012. These projects accounted for about 73 percent of computer time and 50 percent of archival storage used by FES at NERSC in 2012. Some of the projects below have a combined contribution to the case studies that were presented at the review.

NERSC Project ID (Repo)	NERSC Project Title	NERSC Project PI	Review Speaker(s)	Hours Used at NERSC 2012 (M)	Archive Data at NERSC 2012 (TB)	Shared Data on Disk 2013 (TB)
mp19	*Turbulent Transport and Multiscale Gyrokinetic Simulation*	Wei-Li Lee	Jeff Candy, Stephane Ethier, Scott Parker	29.8	34.8	3.0
m1574	*Gyrokinetic Simulations of Multiscale Electron Turbulence for Improved Predictive Modeling of Tokamak Plasmas (ALCC Project)*	Chris Holland	Jeff Candy, Stephane Ethier, Scott Parker			
mp118	*Kinetic simulation of plasma microturbulence, kinetic MHD phenomena and magnetic reconnection processes*	Yang Chen / Scott Parker	Jeff Candy, Stephane Ethier, Scott Parker			
m808	*Energetic Particles: SciDAC GSEP: Gyrokinetic Simulation of Energetic Particle Turbulence and Transport*	Zhihong Lin	Zhihong Lin	17.3	6.9	0.1
m77	*Center for Simulation of Wave-Plasma Interactions: SciDAC Project*	Paul Bonoli	Paul Bonoli	7.2	2.4	0.0
mp2	*LLNL MFE Supercomputing*	Bruce Cohen	C.S. Chang	83.3	14.9	0.5
m499	*Center for Edge Physics Simulation: SciDAC-3 Center*	C. S. Chang	C.S. Chang			
mp288	*3D Extended MHD simulation of fusion plasmas*	Stephen Jardin	Stephen Jardin,	9.3	30.1	5.6
mp200	*Fluid and Hybrid Modeling of Electromagnetic Activity in MFE Plasmas*	Carl Sovinec	Stephen Jardin, Carl Sovinec			
m1255	*Simulation and model validation of the Swarthmore Spheromak Experiment (SSX)*	Vyacheslav Lukin	Vyacheslav Lukin	3.1	5.2	2.6
m172	*Helicity Injected Torus Current Drive and Compact Toroid Studies*	Brian Nelson				
m489	*The Plasma Science and Innovation Center*	Brian Nelson				
m148	*Center for Integrated Computation and Analysis of Reconnection and Turbulence*	Amitava Bhattacharjee	Homa Karimabadi, Kai Germaschewski	16.4	177.7	1.0
m1303	*Petascale Kinetic Simulations in Laboratory and Space Plasmas*	Homa Karima-badi				
mp42	*Simulation of Intense Beams for Heavy-Ion-Fusion Science*	Alex Friedman	Alex Friedman	0.4	0.75	0.3

m74	*Nonlinear Delta-f Particle Simulation of Collective Effects for Heavy Ion Fusion Drivers and High Intensity Particle Accelerators*	Ronald Davidson				
m792	*Study of laser-plasma interactions relevant to direct-drive ICF implosions*	Chuang Ren	Chuang Ren	19.8	79.0	3.0
m412	*Investigations of advanced ignition physics and extreme states of matter*	Warren Mori	Chuang Ren, Frank Tsung			
m1157	*Particle-in-Cell Simulations of Laser Plasma Interactions Relevant to Inertial Fusion Energy*	Frank Tsung	Frank Tsung			
m916	*Ab-initio modeling of the energetics and structure of nanoscale Y-Ti-O cluster precipitates in ferritic alloys*	Brian Wirth	Brian Wirth	2.3	0.085	0
m1200	*Computational modeling of plasma - surface interactions in tungsten exposed to mixed He-H plasmas*					
m1043	*Modeling of tokamak plasmas with large scale instabilities*	Alexei Pankin	Alexei Pankin	0.5	0	0
m649	*Integrated modeling simulations for the plasma edge and core*	Arnold Kritz	Arnold Kritz	0.1	0.3	0
Total of projects represented by case studies				**196**	**352**	**16.1**
All FES at NERSC 2012				**269**	**711**	
Percent of NERSC FES 2012 allocation represented by case studies				**73%**	**50%**	

Draft

6 Findings

6.1 Summary of Requirements

The following is a summary of requirements derived from the case studies.

1. **To meet Fusion Energy Sciences research objectives, FES researchers need computing and data storage resources in excess of those predicted by historical trends.**
    a. Research teams need more than 13 billion hours in 2017, a 50-fold increase over hours used at NERSC in 2012.
    b. Scientists will need to store more than 140 PB of archival data storage in 2017, a huge increase of nearly 200X over 2012.
    c. Collaboratory teams will need 64 petabytes of shared online data in 2017, compared to almost no use of shared data space at NERSC in 2012.
2. **FES codes will require updated mathematical and I/O libraries that will run efficiently on next-generation architectures.**
    a. Threaded linear solvers are needed for many-core systems. The PETSc library is heavily used on today's machines and must either be updated or replaced.
    b. The HDF5, ADIOS, and MPI I/O libraries must be supported on next-generation platforms.
3. **FES scientists need to execute both ensemble runs and large-scale jobs.**
    a. Some problems in FES research require massively parallel computations at the largest possible size to capture the relevant physics over a range of time and length scales.
    b. Scientists need to make many runs of varying sizes and lengths to do parameter studies, validate and verify codes, and perform uncertainty quantification.
    c. Strong and weak scaling are both important.
4. **Teams need effective tools for managing workflows, performing data analysis, and profiling and developing codes.**
    a. Many codes use Python and other software that requires dynamic loading of modules and libraries.
    b. Visualization and analysis tools are needed, including VisIt, Matlab, and AVS-Express.
    c. Optimization tools
5. **Code teams need help porting applications to run well on next-generation architectures.**
    a. Developers need access to training classes, documentation, and online tutorials.
    b. Access to new hardware as early as possible is needed to prepare for future systems.

6.2 Additional Observations

Participants at the meeting made several observations that are not listed in the high-level findings, the most significant of which are listed here.

- **Readiness for next-generation architectures (many-core) varies.** Some groups have working codes that make good use of GPUs, while others are just starting to experiment with porting to many-core processors.

- **Some projects run at low parallel concurrency to conserve allocation.** Since parallel scaling is never perfect, some projects that can sacrifice time-to-solution and execute using limited memory run at a lower parallel concurrency to conserve their allocation of computational time.

- **It is difficult to get access to an entire machine at NERSC.** Other projects that would like, or need, to run full-configuration jobs have difficulty doing so in the NERSC environment.

- **Stable systems and seamless software upgrades are highly valued.**

- **NERSC consulting and account support are also highly valued.**

6.3 Requirements Summary

The following tables list the 2017 computational hours and archival storage needed at NERSC for research represented by the case studies in this report. "Total Scaled Requirement" at the end of the tables represents the hours needed by all 2012 FES NERSC projects if increased by the same factor as that needed by the projects represented by the case studies. The factor of 50 increase is the ratio of the "All FES at NERSC Total Scaled Requirement" to the "All FES at NERSC" 2012 usage from the table in Section 5.2.

Draft

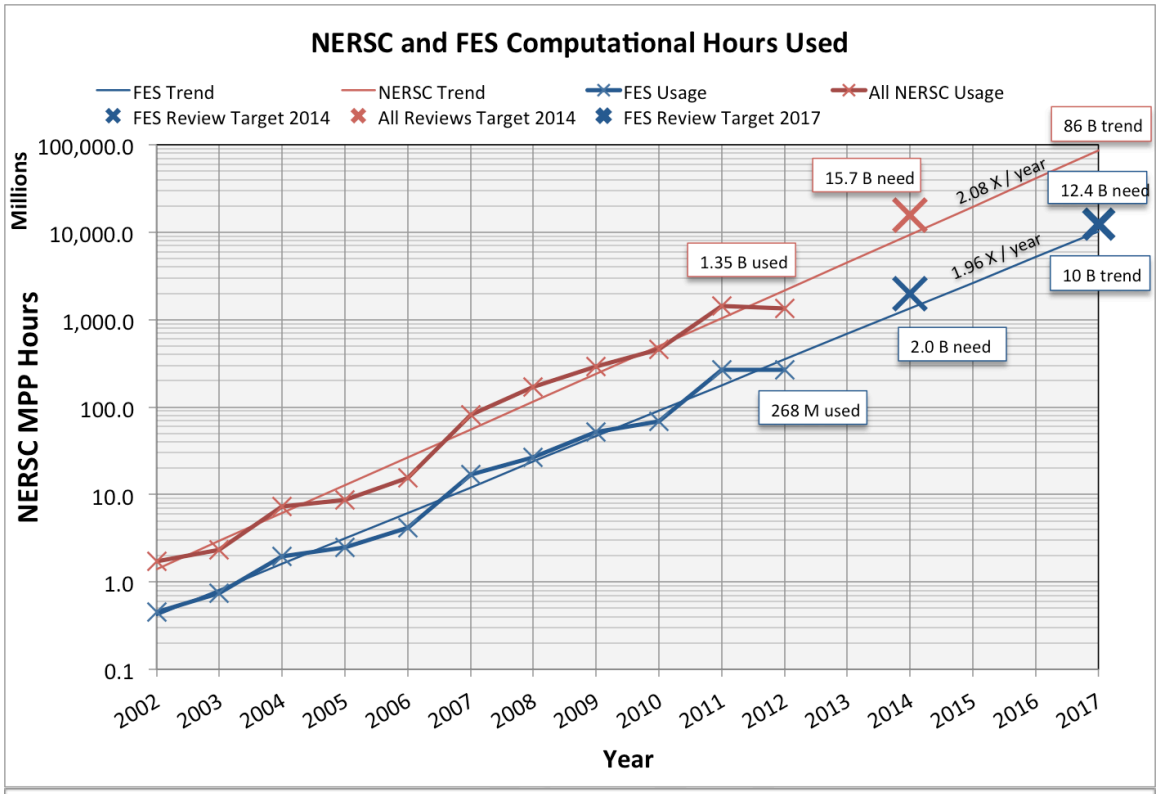
6.3.1 Computing

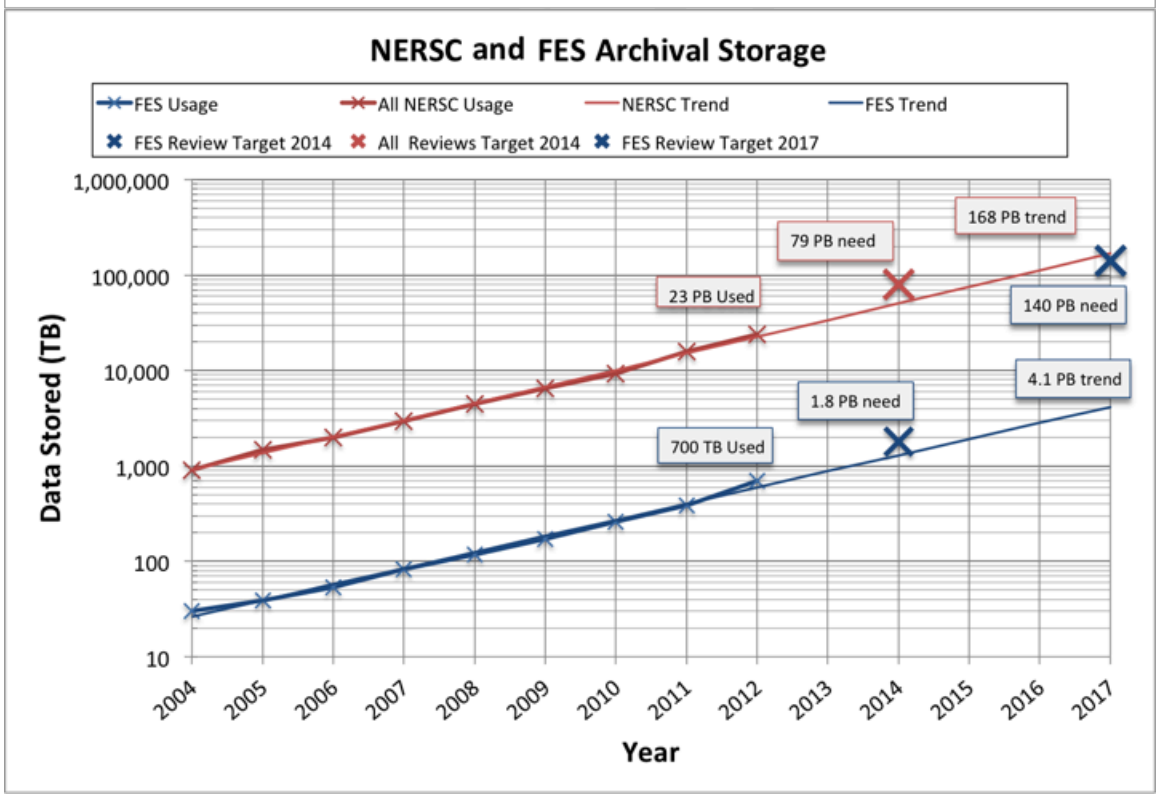
Case Study Title	Repos	PI	Hours Needed in 2017	
			Million Hours	Factor Increase
Transport Calculation and Profile Prediction	m1574, mp19, mp118	Lee, Holland, Chen, Parker	472	16
Energetic Particles: SciDAC GSEP: Gyrokinetic Simulation of Energetic Particle Turbulence and Transport	m808	Lin	1,100	64
Center for Simulation of Wave-Plasma Interactions: SciDAC Project	m77	Bonoli	80	11
Center for Edge Physics Simulation: SciDAC-3 Center	mp2, m499	Cohen, Chang	2,200	26
Study of the Internal Dynamics of ITER	mp288, mp200	Jardin, Sovinec	234	25
Small Scale Plasma Experiments: Simulations and Model Validation	m1255, m172, m489	Lukin, Nelson	200	65
Petascale Kinetic Simulations in Laboratory and Space Plasmas	m148, m1303	Karimabadi, Bhattacharjee	4,147	253
Simulation of Intense Beams for Heavy-Ion-Fusion Science	mp42, m74	Friedman, Davidson	40	100
Study of laser-plasma interactions relevant to direct-drive ICF implosions	m1157, m412, m792	Ren, Mori, Tsung	550	28
Plasma Materials Interactions	m916, m1200, m1709	Wirth	450	200
Integrated Whole-Device Modeling Studies	m1043, m649	Pankin, Kritz	260	430
Total Represented by Case Studies			**9,773**	**50**
Percent of NERSC FES Represented by Case Studies			**73%**	
All FES at NERSC Total Scaled Requirement			**13,332**	**50**

6.3.2 Storage

Case Study Title	Repos	PI	Archival Data Storage Needed in 2017		Shared Online Data Storage Needed in 2017	
			TB	Factor Increase	TB	Factor Increase
Transport Calculation and Profile Prediction	m1574, mp19, mp118	Lee, Holland, Chen, Parker	610	18	60	20
Energetic Particles: SciDAC GSEP: Gyrokinetic Simulation of Energetic Particle Turbulence and Transport	m808	Lin	150	22	100	1,000
Center for Simulation of Wave-Plasma Interactions: SciDAC Project	m77	Bonoli	10	4.1	10	N/A
Center for Edge Physics Simulation: SciDAC-3 Center	mp2, m499	Cohen, Chang	3,000	202	10	20
Study of the Internal Dynamics of ITER	mp288, mp200	Jardin, Sovinec	360	12	20	3.6
Small Scale Plasma Experiments: Simulations and Model Validation	m1255, m172, m489	Lukin, Nelson	200	38	26	10
Petascale Kinetic Simulations in Laboratory and Space Plasmas	m148, m1303	Karimabadi, Bhattacharje e	64,000	360	64,000	64,000
Simulation of Intense Beams for Heavy-Ion-Fusion Science	mp42, m74	Friedman, Davidson	40	53	1	3.3
Study of laser-plasma interactions relevant to direct-drive ICF implosions	m1157, m412, m792	Ren, Mori, Tsung	1,400	18	20	6.7
Plasma Materials Interactions	m916, m1200, m1709	Wirth	300	3,500	100	-
Total Represented by Case Studies			**70,070**	**199**	**64,347**	**3,997**
Percent of NERSC FES Represented by Case Studies			**50**		unknown	unknown
All FES at NERSC Total Scaled Requirement			**140,715**	**199**	unknown	unknown

7 FES and NERSC Trends





8 Magnetic Fusion Energy Science (MFES) Case Studies

8.1 Transport Calculation and Profile Prediction

Principal Investigators: C. Holland[4] Wei-li Lee[3]
Case Study Authors: Jeff Candy[1], Y. Chen[2], S. Ethier[3], C. Holland, S. Parker[2], W.X. Wang[3]
NERSC Repositories: m1574 (C. Holland, PI); mp19 (Wei-li Lee, PPPL, PI); mp118 (Yang Chen, University of Colorado, PI)

8.1.1 Project Description

8.1.1.1 Overview and Context

The overall objective of this research is to better understand the fundamental physics of transport (collisional and turbulent) in tokamaks using a theoretical framework that approximates the solution of the 6D Fokker-Planck-Landau equation. An approximate separation of these equations into collisional (neoclassical) and turbulent (gyrokinetic) components forms the basis of the current approach. Most of the computer time is required by the turbulent component via massively parallel gyrokinetic simulations. The goal is to translate this level of understanding into a predictive modeling capability – which includes design, optimization, and interpretation of future experiments and reactors.

In this report, we cover work based on the codes GTS (PPPL) and TGYRO (General Atomics), but in the final tabulation we also include data for GEM (Univ. of Colorado).

8.1.1.2 Scientific Objectives for 2017

GTS: Experimental validation for NSTX and DIII-D data is our main objective so that we can predict the transport levels in upcoming NSTX-U (upgrade of the NSTX experiment currently underway) and ITER experiments.

TGYRO: Having completed years of validation exercises, and identified regimes where gyrokinetic theory succeeds and fails (so-called L-mode shortfall), we wish to continue development of "sufficiently accurate" models of turbulent transport that can profitably be used in integrated whole-device modeling frameworks. These in turn will be used to design and optimize future experiments. An important point is that agreement isn't perfect, but good enough to provide actual guidance, especially with respect to uncertainty quantification. Specifically, we will continue with detailed validation of gyrokinetic (and reduced gyrofluid) model predictions against experimental observations in U.S. tokamaks (DIII-D, C-Mod, NSTX) to better qualify where current models perform well, and identify parameter regimes where they must be improved. Some attempts to understand aspects of the (near-marginal) core turbulence in ITER are also expected.

---

[1] General Atomics
[2] University of Colorado
[3] PPPL
[4] University of San Diego

8.1.2 Computational Strategies (now and in 2017)

8.1.2.1 Approach

GTS: An important goal right now is the development of a robust algorithm for electromagnetic, finite-beta physics in the GTS code. The more complex field equations greatly increases the time spent in the solver, which is currently implanted with PETSc library routines, which are currently not multi-threaded. Since the rest of GTS is multi-threaded, we are looking at other possible numerical solvers.

TGYRO: We have all the tools for solving the full model hierarchy described in Sec. 1.1: first-principles nonlinear electromagnetic gyrokinetic calculations (GYRO), neoclassical transport (NEO), and a reduced quasi-linear model (TGLF) of the turbulent fluxes. This suite of capabilities is managed by TGYRO to supply the coupling and feedback for production predictive modeling.

8.1.2.2 Codes and Algorithms

GTS: The Gyrokinetic Tokamak Simulation code, GTS, is a global, gyrokinetic particle-in-cell application in general toroidal geometry. There is also an associated neoclassical component, GTC-NEO, that solves a time-dependent form of the neoclassical kinetic equations using a δf particle method.

TGYRO: This application combines GYRO (to solve the nonlinear electromagnetic gyrokinetic equations using Eulerian spectral and finite-difference methods), NEO (to solve for the neoclassical distribution and fluxes/flows with exact linearized Landau collision operator using a spectral expansion scheme in velocity space), and TGLF (as a proxy to GYRO using a fast quasi-linear model that approximates the turbulent transport coefficients). TGYRO itself is a transport manager that couples the above modules to give steady-state profile prediction, as a function of input heating power, for existing devices and future devices.

8.1.3 HPC Resources Used Today

8.1.3.1 Computational Hours

Computational hours used in 2012 include repo m1574 (14M hours used), mp19 (14.6M hours used), and mp118 (1.2M hours used). Several other repositories (gc3, m499, m681, m912) also reported using GEM, totaling about 4.3M hours. The mp118 repository will represent GEM usage in this report. Therefore, the total 2012 usage for this case study is 30 M hours.

8.1.3.2 Parallelism

GTS: We typically use 16,512 cores (2,752 MPI tasks with 6 OpenMP threads/task) for a single run, which allows us to get the most out of our allocation. To code can scale to a larger number of cores but the efficiency of the solver goes down due to the lack of multithreading in PETSc. Weak scaling is more important in order to simulate ITER plasmas.

TGYRO: Typical TGYRO ensemble run with synthetic diagnostics uses 42,240 cores (11 GYRO instances each using 1,280 MPI tasks and 3 OpenMP threads, run for 12 hours, or 506,880 MPP-hours).

8.1.3.3 Data and I/O

NOTE: a single TGYRO ensemble run for m1574 created 55 GB of data. When carrying out many ensemble runs, not all data from each ensemble must be archived. This means that, realistically, less than 1TB of storage will be required per user per year.

Scratch (temporary) space:

5 TB/user (GTS), < 500 GB/user (GYRO)

Permanent (can be shared, NERSC project Global File system):

20 TB / 4 TB (GTS), < 1 TB/user (GYRO)

HPSS permanent archival storage:

60 TB (GTS), < 1 TB (GYRO)

There is nothing out of the ordinary in our usage of these three spaces: the scratch file system is where we run all of our simulations. The project area has greatly facilitated data sharing. We share codes and data with the other members of our project through the project space. We archive important simulation data in HPSS. We sometimes need to share data between Hopper, Carver, Edison, and the DTN systems. Here is how our codes perform I/O:

GTS: Some parallel I/O with ADIOS library calls for large data sets and checkpoint-restart, as well as single-core FORTRAN ASCII I/O.

TGYRO: Large restart files are written via MPI-IO, simulation data written in ASCII, option for HDF5 output, but not generally favored by users.

8.1.4 HPC Requirements in 2017

8.1.4.1 Computational Hours Needed

TGYRO: We will identify the range of scales needed for modeling burning plasma conditions. A very important question: Is ITG-scale physics enough, or is multiscale required? This and other key questions to be answered in near term will impact our future HPC requirements. Also, will we want to move towards production level simulations and ensemble UQ; how big will each ensemble element need to be? Plausible 2017 predictive workflow: improved version of TGLF or some other model as workhorse, with GYRO "spot checks" at periodic and critical points.

GTS: We expect our project to need between 200 M to 300 M core-hours during CY 2017 for ITER simulations with finite-beta physics.

TGYRO: Probably about 100M core hours. Number of Compute Cores

8.1.4.2 Parallelism

GTS: We have already demonstrated with another gyrokinetic PIC code, GTCP, that we can achieve very high scalability (> 700,000 cores) by adding a radial domain decomposition along with optimized multi-threading and data layout. The same optimizations can be implemented in GTS in order to improve efficiency and scalability.

TGYRO: We expect to be using GPUs in 2017, and effort is underway now to develop this capability on Titan at ORNL. We believe that the GYRO/TGYRO framework is now maximally parallelized using MPI and OpenMP, with scalability beyond 100K cores if required. By 2017

we expect to also solving more complex 6D equations, and we cannot make projections about that for 2017.

8.1.4.3 Data I/O

GTS: ITER data sets will be much larger so the I/O numbers should increase by about an order of magnitude.

TGYRO: We do not expect data needs to increase significantly per user – perhaps by a factor of 2 or 3. However, in the future we envision an increasing number of TGYRO users.

8.1.4.4 Memory Required

GTS: 16 Gbytes per node is a reasonable number although more is better. 320 Tbytes of memory in total will be needed for large ITER simulations.

TGYRO: 16 Gb/node is acceptable. Probably less than 100TB aggregate.

8.1.4.5 Many-Core and/or GPU Architectures

Our codes use OpenMP multi-threading and we are actively working on a GPU port of key routines to take advantage of this specialized architecture for both GTS and GYRO.

8.1.4.6 Software Applications and Tools

GTS: Unless we find something better we will still use the PETSc library to implement our solvers, as well as a random number generator (SPRNG) and spline routines (PSPLINE). Our I/O is implemented with ADIOS.

TGYRO: netCDF, multithreaded fftw3, BLAS, LAPACK, mumps/superlu, HDF5.

8.1.4.7 HPC Services

We will need everything that NERSC provides, including consulting or account support, data analytics and visualization, training, collaboration tools, web interfaces, federated authentication services, gateways, etc.

8.1.4.8 Scientific Achievements with 32X Current Resources

One could probably achieve as much by reducing bureaucracy and barriers to resource usage as could be by adding more computational resources. The ease-of-access to NERSC resources is a current strength in this author's opinion and should be maintained. Moreover, true scientific advances require improvements on 2 fronts: (1) better coupling and workflow for predictive simulations of the type achievable today, (2) an improved theoretical formulation that can treat the pedestal and separatrix correctly – using perhaps the original 6D FPL equation. This is a new area that goes beyond traditional gyrokinetics. There is no accepted approach and no codes to solve this problem. By 2017, we suspect there will be. To reiterate, progress in (2) is a matter of progress in theoretical physics much apart from access to resources.

8.1.4.9 Requirements Summary

GTS (mp19)	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (Million)	14.6	300
Scratch storage		100 TB
Shared global storage (project)	1.3 TB	
Archival storage (HPSS)	32 TB	600 TB
Number of cores used for prod. runs	16,512	160,512
Memory per node	32 GB	32 GB
Aggregate memory	22 TB	220 TB
TGyro (m1574)	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (Million)	14 M	150 M
Scratch storage		10 TB
Shared global storage (project)	1.7 TB	10 TB
Archival storage (HPSS)	0.9 TB	10 TB
Number of cores used for prod. runs	20K	100K
Memory per node	16 GB	16 GB
Aggregate memory	10 TB	100 TB
GEM (mp118)	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (Million)	1.24	22 M
Scratch storage		
Shared global storage (project)	0	50 TB
Archival storage (HPSS) Archival bandwidth	1.2 TB 0.1-1 Gbit/s	 1-10 Gbit/s
Number of cores used for prod. runs	10K	100K
Memory per node	16 GB	16 GB
Aggregate memory	10 TB	100 TB

Draft

8.2 Energetic Particles and the SciDAC Gyrokinetic Simulation of Energetic Particle Turbulence and Transport Project

Principal Investigator: Zhihong Lin (University of California, Irvine)
NERSC Repository: m808
Case Study Coauthor: Scott Parker (University of Colorado)

8.2.1 Project Description

8.2.1.1 Overview and Context

Energetic particles produced by fusion reactions and auxiliary heating can excite various instabilities in fusion experiments such as ITER. Associated nonlinear wave-particle interactions can generate significantly enhanced levels of energetic particle transport that would degrade overall plasma confinement and damage fusion devices.

The fully self-consistent simulation of energetic particle turbulence and transport must incorporate three physics elements: kinetic effects of thermal particles at the microscale, nonlinear interactions of a large number of meso-scale energetic particle instabilities, and cross-scale couplings of macro-meso-micro turbulence. The large dynamical ranges of spatial-temporal processes further require global simulation codes that are efficient in utilizing massively parallel computers at the petascale level and beyond.

In the current efforts, we develop gyrokinetic turbulence simulation as a new paradigm for a predictive capability of the energetic particle confinement in strongly self-heated burning plasmas such as ITER, and propose to extend the capability of highly scalable gyrokinetic turbulence codes based on the complementary particle-in-cell (PIC) and continuum codes to study the energetic particle driven instabilities and their interaction with background thermal plasmas. This case study focuses on the PIC codes GTC and GEM.

8.2.1.2 Scientific Objectives for 2017

Confinement and stability properties of fusion plasmas depend on nonlinear interaction of multiple physical processes covering disparate spatial and temporal scales. The current effort is to initiate the development of the capability for a first-principles simulation providing long time scale macroscopic dynamics with the fidelity of a microscopic kinetic simulation.

Our ultimate goal is to build the predictive capability for energetic particle turbulence and transport in the ITER burning plasmas, which requires understanding nonlinear physics of energetic particle instability, predicting energetic particle transport given a fixed energetic particle drive, and self-consistent gyrokinetic simulations of a full burst cycle of energetic particle turbulence.

8.2.2 Computational Strategies

8.2.2.1 Approach

In the gyrokinetic simulation, the phase attribute of the fast gyration (or cyclotron) motion of the charged particles around the magnetic field lines is averaged away, reducing the dimensionality of the system from 6D to 5D phase space. This method removes the fast

cyclotron motion, which has a much higher frequency than the characteristic waves of plasma microturbulence.

The particle-in-cell method consists of moving particles along the characteristics of the gyrokinetic equation. The electromagnetic fields are obtained by solving the Poisson equation and Ampere's law on a spatial mesh after gathering the charge density on the grids. The electromagnetic forces are subsequently scattered back to the particle positions for advancing the particle orbits. The use of spatial grids and the procedure of gyroaveraging reduce the intensity of small-scale fluctuations (particle noise). Particle collisions can be recovered as a "subgrid" phenomenon via Monte Carlo methods. Using a perturbative method, where only the perturbed distribution function is calculated in the simulation, further reduces the particle noise.

8.2.2.2 Codes and Algorithms

The global gyrokinetic toroidal code (GTC) is a massively parallel particle-in-cell code for first-principles, integrated simulations of the burning plasma experiments such as ITER. GTC is the key production code for the U.S. DOE SciDAC project, GSEP Center, and the National Special Research Program of China for ITER. A single version of GTC is currently capable of both full-f and δf simulations, gyrokinetic or fully kinetic ions, kinetic electrons and electromagnetic fluctuations, general toroidal geometry with shaped, up-down asymmetric equilibrium and experimental plasma profiles, equilibrium current for kink drive, multiple ion species, neoclassical effects with Fokker-Planck collision operators conserving particle, momentum and energy, equilibrium radial electric field with toroidal and poloidal rotations, and sources/sinks and external antenna. GTC webpage: http://phoenix.ps.uci.edu/GTC/

GEM is a comprehensive electromagnetic gyrokinetic δf particle code that includes the full dynamics of gyrokinetic ions and drift-kinetic electrons. The simulation is useful for studying well-magnetized plasma physics and is especially powerful because it is accurate at very-low fluctuation levels. Electron-ion collisions are included as well as the full-capability to model general axisymmetric toroidal equilibria. The effects of impurity ions and the equilibrium radial electric field are included in GEM. GEM is also used to model energetic particle driven MHD instabilities using a slowing down distribution for the energetic component. GEM webpages:

http://en.wikipedia.org/wiki/Gyrokinetic_ElectroMagnetic

http://cips.colorado.edu/Group/Simulation/Gem.php

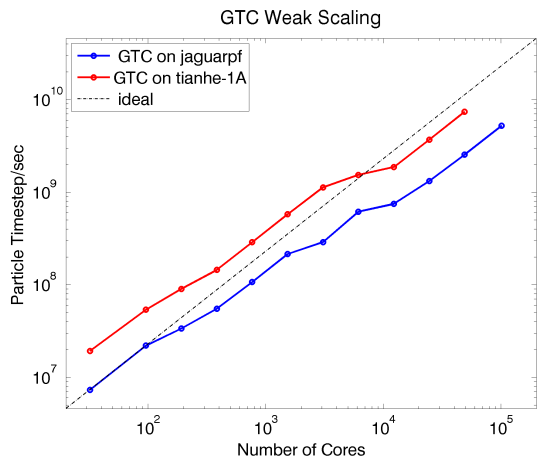
8.2.3 HPC Resources Used Today

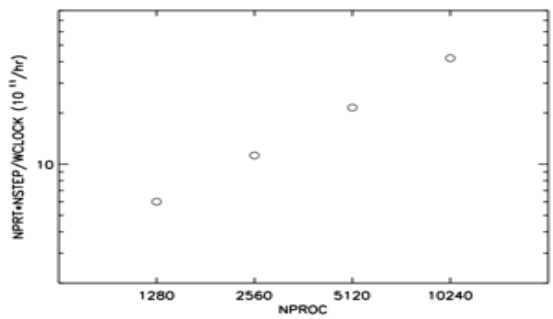
8.2.3.1 Computational Hours

In 2012, GTC used about 25M hours on Hopper at NERSC, 20M hours on Jaguar and Titan at ORNL, and 5M hours of Tianhe-1A at the National Supercomputing Center in Tianjin, China.

8.2.3.2 Parallelism

GTC is a platform-independent program using standard FORTRAN, message passing interface, and OpenMP, and achieves nearly perfect scalability on massively parallel computers, including the world fastest computer with more than 10^5 cores and graphic processing unit (GPU) acceleration. It has recently been ported and optimized for the new fastest supercomputer using the many integrated core (MIC) architecture.





GEM Scaling on Titan

8.2.3.3 Data and I/O

Scratch space used for restart data of active simulations: 10TB. Permanent space used for analysis of data from finished simulations: 10TB. HPSS, if used, for data from selected simulations that may be analyzed again in the future: 7 TB

8.2.4 HPC Requirements in 2017

8.2.4.1 Computational Hours Needed

GTC needs 1 billion hours in 2017 for longer time scale simulations with higher resolutions (e.g., large number of grids and particles).

GEM needs 100 million hours in 2017.

8.2.4.2 Parallelism

GTC could use 1,000,000 cores in 2017.

8.2.4.3 Data and I/O

Data size increases with core numbers.

8.2.4.4 Scientific Achievements with 32X Current Resources

With access to 2X more resources each year we could do first-principles, integrated simulation of cross-scale interaction between multiple physics processes, e.g., coupling between microturbulence, energetic particle, magnetohydrodynamics (MHD), and/or neoclassical transport. These couplings play a key role in the excitation and evolution of the performance-limiting and integrity-threatening MHD instabilities in the burning plasmas, e.g., neoclassical tearing mode, resistive wall mode, sawtooth, etc..

8.2.4.5 Memory

Memory stays unchanged per core, and increases with cores per node globally.

8.2.4.6 Many-Core and/or GPU Architectures

GTC has been ported and optimized for GPU and MIC, respectively.

Draft

8.2.4.7 Software Applications and Tools

GTC needs linear solvers (e.g., PETSc), I/O (e.g., ADIOS)

8.2.4.8 HPC Services

We need help in consulting and account support, training, and collaboration tools from NERSC.

8.2.4.9 Time to Solution and Throughput

Fast turnaround time for small jobs would be very helpful.

8.2.4.10 Data Intensive Needs

The biggest dataset from GTC is the restart data.

8.2.4.11 Requirements Summary

Requirements for GTC	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours (million)	11.6	1000
Scratch storage	10 TB	100 TB
Shared global storage (/project)	0.1 TB	100 TB
Archival storage (HPSS)	4.9 TB	100 TB
Number of cores* used for production runs	10,000	1,000,000
Memory per node	10 GB	100 GB
Aggregate memory	10 TB	1000 TB
Requirements for GEM	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (million)	5.6	100
Scratch storage	5-10 GB	500 GB
Shared global storage (/project)		
Archival storage (HPSS)	2 TB	50 TB
Number of cores* used for production runs	10k	100k (+GPU)
Memory per core	1 GB	1 GB
Memory per GPU		16 GB[1]

[1] Total GPU memory ideally should equal total node memory, multi-GPU per node is advantageous.

8.3 Center for Simulation of Wave-Plasma Interactions (CSWPI)

Principal Investigator: Paul Bonoli (MIT)
NERSC Repository: m77

8.3.1 Project Description

8.3.1.1 Overview and Context

The over-arching goal of the project is to quantitatively understand how high power (tens of MW) radio frequency (RF) power in the ion cyclotron frequency range of frequencies (ICRF) and in the lower hybrid range of frequencies (LHRF) propagates from an external antenna and how it is subsequently absorbed in a tokamak plasma. This capability is needed to understand how to optimally use this power to heat, drive current, control plasma profiles, control plasma stability, and avoid parasitic losses in magnetically confined fusion plasmas including ITER. This problem is computationally intensive because of the non-linear, 3D, multi-scale characteristics of the problem. Present three-dimensional (3-D) runs for linear and quasi-linear models of the plasma core can take 100-200 K processor hours on today's machines. Three-dimensional non-linear runs that couple the core and edge (including the antenna) are expected to take 10-1000 times more cycles.

8.3.1.2 Scientific Objectives for 2017

Research carried out by the SciDAC Center for Simulation of Wave-Plasma Interactions (CSWPI) has the goal of developing the first-ever integrated edge-to-core simulation capability for the coupling, propagation, and absorption of RF waves. The Center's research is organized into four major thrusts:

(1) Coupled core-to-edge simulations that will lead to an increased understanding of parasitic losses of the applied RF power in the boundary plasma between the RF antenna and the core plasma. In existing experiments with modest power levels and short time durations, 10% or more of the applied RF power can be lost in the edge regions or in the launcher. At the power levels and pulse lengths anticipated for ITER, this level of losses would be especially problematic.

(2) Development of models for core interactions of RF waves with energetic electrons and ions that include a more accurate representation of the particle dynamics in the combined equilibrium and wave fields, thus providing an understanding of how these energetic species affect power flow in the confined plasma.

(3) High-resolution simulations of RF effects on fast-particle driven instabilities driven by fusion alpha particles or NBI ions that will quantify if these interactions increase (decrease) the instability drive that can lead to reduced fusion power.

(4) Development of improved algorithms that will take advantage of massively parallel computing platforms up to the multi petascale level and beyond to exascale, in order to achieve the needed physics, resolution, and/or statistics to address these issues. This research requires an extensive program of code verification and validation to insure that critical wave-plasma interactions are accurately reproduced in the simulations.

Draft

The development of a core-to-edge integrated simulation code that will be capable of predicting the absolute amount of power absorbed in both the edge and good confinements regions of the discharge will provide essential support for the ongoing experiments on NSTX, DIII-D and Alcator C-Mod, and will have a critical impact on the design of the ICRF system and operational scenarios for ITER. It will also provide a foundation for computing the macroscopic and microscopic source information needed by MHD and transport codes in fully integrated modeling activities such as the Proto-type Fusion Simulation Projects and the Fusion Simulation Project (FSP) itself.

These objectives reflect the five-year goals of the CSWPI, which would include the last 2.5 years of the project's current funding phase, which ends in 2015, and assumes the project receives continuation funding for years beyond 2015.

8.3.2 Computational Strategies (now and in 2017)

8.3.2.1 Approach

The computational problem consists of solving the wave equation in the ICRF and LHRF regimes, subject to boundary conditions that represent RF fields imposed by either an ICRF antenna structure or a LHRF waveguide launching structure. Modification of the electron and ion distribution functions by the RF power is calculated by solving a Fokker Planck equation, where the effect of the RF power is quantified through either a quasi-linear diffusion operator or an RF "kick" operator. RF-induced changes in the distribution function are incorporated in the full-wave solvers by evaluating the plasma response in the solvers using the modified particle distribution. The interaction between the wave solvers and Fokker Planck codes is carried out in time through an explicit iteration strategy. The self-consistent effect of the ICRF antenna launching structure is now being included in this problem by iterating an antenna code with the wave solver. Coupling between the wave solver and antenna code is iterative (see next section).

8.3.2.2 Codes and Algorithms

Three classes of codes are used for addressing the CSWPI physics objectives. First, full-wave codes solve the integral-differential Maxwell's equations with a non-local, linear RF plasma conductively to calculate the RF electric fields that are driven by external power sources. The All Orders Spectral Algorithm (AORSA) and the Toroidal Ion Cyclotron Code (TORIC) are the principal codes used by the project. The AORSA solver is fully spectral and employs a Cartesian coordinate system while the TORIC code is semi-spectral with Fourier modes in the poloidal and toroidal directions and finite element in the radial direction. The number of spectral modes required to achieve numerical convergence in these solvers is dictated by the shortest wavelength in the system that must be resolved. The plasma response can either be for thermal plasma components with a Maxwellian distribution function or non-thermal components with a specified distribution function. Typical problems require solving for 10^5 to 10^6 unknowns for a single toroidal mode coupled by an ICRF launching structure. For lower hybrid heating and current drive calculations using TORLH, ~10 times more resolution is required, with an approximate 100 fold increase in the number of unknowns.

The RF fields can in turn modify the distribution function. The second class of code models the evolution in time and phase space (velocity and configuration) of the distribution function as determined by quasi-linear RF interactions. Explicit code coupling is valid for this problem because the RF time scale, nanoseconds, is much shorter that the distribution function evolution that takes place on transport/collisional time scales, 10s of ms to

fractions of a second. Both continuum (CQL3D) and Monte Carlo (ORBIT-RF and sMC) techniques are employed. Although the interaction between the wave solvers and Fokker Planck / Monte Carlo codes is highly nonlinear, the problem is solved stably by iterating between the physics modules as time is evolved on the quasi-linear time scale.

The RF fields coupled by the ICRF launching structure or LHRF waveguide are simulated using the massively parallel, multi-physics, kinetic electromagnetics code VORPAL, developed by the Tech-X Corporation. Only the finite-difference-time-domain (FDTD) capabilities in VORPAL are utilized for these simulations, which includes a cold plasma, accurate antenna geometry representation and a nonlinear RF sheath boundary condition.

Thus five different physical models are iteratively coupled: (1) the plasma conductivity for non-Maxwellian distribution functions, (2) a wave solver incorporating this non-Maxwellian conductivity, (3) the quasi-linear operator for the non-thermal distribution, (4) a Fokker-Planck solver or Monte Carlo code, and (5) an antenna coupling code to simulate the RF fields in vacuum.

For the full wave AORSA solver, a completely dense matrix results from the integral plasma conductivity relation and spectral basis set used in that code. A block tri-diagonal system with completely dense blocks results from the algebraic conductivity (finite ion Larmor radius expansion) and spectral basis set used in the TORIC solver. Factoring these matrices is often the dominant computational effort, and we rely on dense solvers (ScaLAPACK or HPL) and either Thomas or cyclic reduction techniques for AORSA and TORIC/TORLH respectively. Filling the matrices and post processing can take a significant fraction of the total computational time, especially for non-thermal distributions. Efficient integration/quadrature techniques and special function evaluations are critical for these steps. Both of the full wave codes solve for one toroidal mode at a time, and must be run once for each of the 50-100 significant toroidal modes of the antenna. The final plasma response is computed by linear superposition of the electric field solutions of the individual toroidal modes. Recently we have successfully used a computational framework capability (the Integrated Plasma Simulator – IPS) developed through the SWIM Proto-typo Fusion Simulation Project in order to simultaneously simulate individual toroidal modes through multiple instantiations of the TORIC ICRF solver. This capability is expected in the future to make 3-D time dependent simulations of ICRF heating routine.

The Fokker-Planck solvers both rely on particle-based techniques (for generating the quasilinear operator for CQL3D) and for determining a Monte-Carlo solution for ORBIT-RF and sMC. In order to evaluate the conductivity operator in the full-wave solvers using the statistical particle lists from the Monte Carlo codes, we have developed an algorithm that transforms these particle lists into continuum distribution functions – a code named "p2f".

The FDTD algorithm employed in the VORPAL code is known to scale well due to only requiring the communication of local domain boundaries to neighboring nodes. However, the work required scales as N4 for 3 (spatial) + 1 (time) dimensions, with the memory required being 30 double precision variables at each spatial cell, with ~ 96 million (M) cells per current strap for an ICH antenna giving ~ 23 GB per strap and requiring ~ 16,000 CPU hours. An algorithm for coupling between the full-wave solver and antenna code is now being developed based on the classic Alternating Schwarz method of domain decomposition and MPE (Minimum Polynomial Extrapolation) vector extrapolation iterative technique.

8.3.3 HPC Resources Used Today

8.3.3.1 Computational Hours

During 2012 the CSWPI used 7,717,153 MPP hours through an ERCAP Request at NERSC. Dr. David Green in our project also received a 2012 Discretionary Award of 3,200,000 hours on the CRAY XK6 at ORNL for preparation of an INCITE Proposal.

In addition to our ERCAP allocation at NERSC for 2013, Dr. David Green in the CSWPI has been awarded 2,000,000 hours on the TITAN supercomputer at the Oak Ridge Leadership Class Facility (OLCF). The allocation will be used to demonstrate coupled core to edge ICRF simulations using the AORSA wave solver and VORPAL antenna code, taking advantage of the combined CPU/GPU architecture on TITAN.

8.3.3.2 Parallelism

(1) The number of compute cores used in production runs depends on the physics problem that is being simulated. The typical number of cores that we use in production runs is:

ICRF minority heating (single toroidal mode): TORIC (64 cores), AORSA (512 cores)

ICRF Mode conversion (single toroidal mode): TORIC (255-512 cores), AORSA (10,000-50,000 cores)

LHRF Simulations (single toroidal mode) : TORLH (1000-8000 cores)

Fokker Planck Simulations: CQL3D (24-48 cores), ORBIT RF, sMC (~10,000 cores)

Antenna Simulations: VORPAL (4000-30,000 cores)

(2) Weak scaling is important to us: Recently we have started to take advantage of the fact that each toroidal mode simulation is independent, so it is possible to carry out the toroidal mode simulations concurrently. We are using a lightweight python framework called the Integrated Plasma Simulator (IPS) to prepare, perform, and process the concurrent toroidal mode simulations. In these simulations for example, we could perform 40 simulations simultaneously using 256 cores per toroidal mode simulation (assuming TORIC and ICRF mode conversion), requiring a total allocation of 10,240 cores. We could also perform 40 toroidal mode simulations using 2048 cores per toroidal mode (assuming AORSA and ICRF mode conversion), requiring a total of 81,920 cores. In either case, both TORIC and AORSA achieve close to perfect weak scaling.

(3) Strong scaling is also important to us: Simulations of LHRF waves and ICRF mode conversion requires more cores because very short wavelength modes must be resolved, requiring higher mode resolution, resulting in a larger problem size. In these cases it is necessary to distribute the problem across more nodes and cores in order to be able to fit the problem in machine memory, while at the same time maintaining scalability with increasing processor count.

8.3.3.3 Data and I/O

Scratch (temporary) space: 5 TB. Used to run large simulations and temporarily store results until they can be transferred to HPSS or to local computers.

Draft

Permanent (can be shared, NERSC Global Filesystem /project): None. NERSC Global Filesystem is used by project members for code development and for low demand archiving of project codes and simulation results. The CSWPI SciDAC Center does not use /project space but it is our intention to utilize this resource in the future because our project does rely on sharing codes and results among members of the Center.

HPSS permanent archival storage: 5 TB. Used by some project members. Most simulation results are transferred back to local computers via scp and several of our codes (such as the TORIC and CQL3D codes) are under SVN control.

We do not usually share data between machines at NERSC, however during 2013 we anticipate needing to do some minimal data sharing between Hopper and Edison.

Typically input is done using small (in size) ASCII data files. Wave field solutions, RF absorption profiles, and 4-D quasilinear diffusion coefficients and RF kick operators are stored in NetCDF files.

At times we have experienced code failures because of problems with the NFS File System (for example "Stale NFS file").

8.3.4 HPC Requirements in 2017

8.3.4.1 Computational Hours Needed

We will require about 60M core-hours in CY2017 to achieve goals in Section 8.3.1.2. The primary factor driving the requirement is the need to use 3-D fields from the AORSA wave solver when coupling to the VORPAL antenna code in the coupled core to edge simulations of ICRF heating.

8.3.4.2 Parallelism

(1) In 2017 the number of conventional compute cores that we expect to be using for single toroidal mode simulations is a bit higher than the upper limits that were given in Section 8.3.2.5:

> ICRF minority heating (single toroidal mode): TORIC (128 cores), AORSA (1024 cores)
>
> ICRF Mode conversion (single toroidal mode): TORIC (512-1024 cores), AORSA (10,000-50,000 cores)
>
> LHRF Simulations (single toroidal mode): TORLH (4000-8000 cores)
>
> Fokker Planck Simulations: CQL3D (24-128 cores), ORBIT RF, sMC (~10,000 cores)
>
> Antenna Simulations: VORPAL (4000-30,000 cores)

The increased number of cores for CQL3D assumes that an implicit matrix solve will have been implemented in the code for 3-D ($v_\perp$, $v_{//}$, r) simulations.

(2) In 2017 we expect to be doing 3-D simulations of ICRF minority heating and mode conversion, and simulations of LHRF waves where each toroidal mode is run concurrently. These types of simulations might typically use:

ICRF minority heating with TORIC: (40 toroidal modes) × (128 cores / mode) = 5120 cores.

ICRF minority heating with AORSA: (40 toroidal modes) × (512 cores / mode) = 20480 cores.

ICRF mode conversion with TORIC: (40 toroidal modes) × (512 cores / mode) = 20,480 cores.

ICRF mode conversion with AORSA: (40 toroidal modes) × (2048 cores / mode) = 81,920 cores.

ICRF mode conversion with AORSA: (40 toroidal modes) × (

LHRF simulations with TORLH: (60 toroidal modes) × (1024 cores / mode) = 61440 cores.

ICRF coupling simulations with AORSA + VORPAL will require 300-2000 toroidal modes using CPU + GPU architectures (see Section 1.3.6).

8.3.4.3 Data and I/O

Our project will have only small data storage requirements. The AORSA-VORPAL coupling represents the single largest storage requirement generated by our project. We estimate ~ 100 GBytes per 3-D AORSA-VORPAL run, giving ~ 500 GBytes that can easily be transferred off site. Consequently our data storage requirements in 2017 will not change drastically from 2012, with the exception of a new /project area:

Scratch space: ~ 10 TBytes

Permanent: ~ 5 TBytes for /project area

HPSS permanent archival storage: 10 TB

8.3.4.4 Scientific Achievements with 32X Current Resources

The 60M core-hours needed by our project in 2017 is approximately 8X our 2012 usage of ~8M hours. This estimate was based on only a limited set of five production runs consisting of three validation scenario production runs for NSTX-U, DIII-D, and ITER, and ideally two further productions to investigate possible parasitic loss mitigation strategies. If we had access to 2X more resources each year, this would enable us to perform double the number of validation scenario production runs and to perform 2-3 x more production runs that would investigate parasitic loss mitigation strategies for ITER.

8.3.4.5 Memory

The largest memory requirements come from our coupled core to edge simulations using AORSA+VORPAL. We plan to do production type simulations for three devices (NSTX, DIII-D, and ITER). Below are the memory requirements for each of these problems:

NSTX: 257 × 513 (R, Z) grid

301 toroidal modes

Aggregate memory requirement (all toroidal modes): 692 Tbyte

Per node memory requirement: 27 GB / node, assuming 2040 cores per toroidal mode with 24 cores per compute node.

DIII-D: 257 × 513 (R, Z) grid

601 toroidal modes

Aggregate memory requirement: 1,382 Tbyte

Per node memory requirement: 27 GB / nodes, assuming 2040 cores per toroidal mode with 24 cores per compute node.

ITER: 513 × 513 (R, Z) grid

2001 toroidal modes

Aggregate memory requirement: 18,209 Tbyte

Per node memory requirement: 23 GB / nodes assuming 10,200 cores per toroidal mode with 24 cores per node.

8.3.4.6 Many-Core and/or GPU Architectures

We currently have a focused effort in the CSWPI SciDAC Center to take advantage of the hybrid CPU/GPU architecture on the TITAN supercomputer at OLCF in order to carry out our coupled core to edge ICRF simulations using the AORSA solver and the VORPAL antenna code. Drs. David Green and David Smithe in our Center have submitted a detailed proposal to acquire resources on TITAN to the ASCR Leadership Computing Challenge (ALCC) Program. The proposal is titled "Unraveling the Coupling of Radio Frequency Power to Fusion Plasmas". The core to edge coupling problem with AORSA + VORPAL is not even possible unless one uses a platform with a CPU/GPU architecture.

We have begun development on the Titan GPU based architecture for the matrix factorization in the AORSA solver, based on recent development work on an out-of-core (OOC) LU factorization. The OOCLU library implements a parallel LU factorization algorithm for large dense complex matrices that takes advantage of GPU acceleration. The library is designed to be compatible with the ScaLAPACK LU factorization routine PxGETRF. An external memory (or out-of-core) left-looking algorithm allows significant problems that are larger than available GPU device memory to be factored.

VORPAL has also been ported to take advantage of GPU acceleration. The GPU architecture is particularly well suited to the FDTD algorithm with strong scaling retained on the Dirac GPU machine at NERSC. Specific challenges for porting VORPAL to a heterogeneous architecture were the creation of dynamically generated kernels for initial/boundary conditions supplied at runtime (using a code generator solution), and efficient hiding of the GPU-to-CPU and CPU-to-CPU data transfer latency which required reordering the execution steps.

The Table below has been reproduced from the ASCR Leadership Computing Challenge (ALCC) Proposal submitted by Dr. David Green in our Center and summarizes the memory and flop requirements that will be needed to perform the AORSA+VORPAL coupling calculation on the TITAN CPU/GPU device. Notice the NSTX simulation can be done with

CPUs only, the DIII-D simulation can be done with either CPUs or GPUs, and the ITER simulation can only be done with GPUs.

Device	2-D Grid	No. Toroidal Modes	Titan Core Hours (M)	Memory (TB)	Assumed Gflops per node
NSTX	257 x 513	301	0.007 x 301 x 2 x 30/16= 7.8	2.3 x 301 = 692	6.7 x 8 = **53.6** (CPU only with 8 MPI tasks)
			0.009 x 301 x 30/16 = 5.4		4.7 x 16 = **75.2** (CPU only with 16 MPI tasks)
DIII-D	257 x 513	601	0.009 x 601 x 30/16 = 10.7	2.3 x 601 = 1,382	**75.2** (CPU only with 16 MPI tasks)
			0.005 x 601 x 30/16 = 6.0		8.0 x 16 = **128.0** (CPU complex*8 PGESVR with 16 MPI tasks)
			0.0038 x 601 x 30/14 = 5.0		**155** (GPU only complex*16 pzgetrf_ooc2)
			0.0024 x 601 x 30/14 = 3.0		**250** (GPU only complex*8 PGESVR)
ITER	513 x 513	2001	0.03 x 2001 x 30/14 = 131.6	9.1 x 2001 = 18,209	**155** (GPU only complex*16 pzgetrf_ooc2)
			0.019 x 2001 x 30/14 = 81.6		**250** (GPU only complex*8 PGESVR)

8.3.4.7 Software Applications and Tools

Our project makes use of IDL, MATLAB, pgplot, python, and ViSiT for post-processing and data visualization. We also use python for code management, especially when we are performing time dependent integrated simulations with our wave solvers and Fokker Planck codes.

We employ netCDF and HDF5 for I/O.

8.3.4.8 HPC Services

I expect the CSWPI SciDAC Center will continue to need both consulting services and account support.

8.3.4.9 Time to Solution and Throughput

In 2017 we will continue to need fast turnaround several times a year coinciding with the 4-6 weeks before major meetings in Fusion Energy Sciences:

APS-Division of Plasma Physics Meeting (held in early to mid-November, yearly)

Sherwood Theory Conference (held in mid-April, yearly)

IAEA Fusion Energy Conference (held in mid-October, every other year – 2013, 2015, and 2017)

Conference on RF Applications is Plasmas (held in May-June, every other year – 2013, 2015, and 2017).

8.3.4.10 Data Intensive Needs

We do not think we will have additional needs regarding data transfer. The data from our individual codes is small enough to be transferred off-site to local computers. Data from time dependent integrated simulations performed using the SWIM framework can be stored efficiently in a /project area or in HPSS.

8.3.4.11 What Else?

The features of an HPC system that are most important to our Center are stability, seamlessness in software upgrades, responsiveness from the consulting staff to problems encountered by users, and fast turnaround times.

In this regard, one of the most pressing concerns we continue to have is the 2-3 day queue wait that occurs on Hopper. At times we have been forced to "daisy-chain" long jobs together in the debug queue in order to perform development work in preparation for production type runs – a practice that is frowned upon by NERSC.

8.3.4.12 Requirements Summary

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours (million)	7.2	80
Scratch storage	4 TB	10 TB
Scratch bandwidth	Negligible	Negligible
Shared global storage (/project)	0 TB	10 TB
Archival storage (HPSS)	2.5 TB	10 TB
Archival bandwidth (HPSS)	Negligible	Negligible
Number of cores* used for production runs	2,000-10,000 (no parallel concurrency for toroidal modes)	600,000-3,000,000 (assumes ~ 300 toroidal modes simulated concurrently).
Memory per node	20-25 GB	20-25 GB
Aggregate memory	~100-1,000 TB	1,000-20,000 TB

Draft

8.4 Simulation of Plasma Edge Physics

Principal Investigator: C.S. Chang (Princeton Plasma Physics Laboratory)
Case Study Contributor: X. Xu (Lawrence Livermore National Laboratory)
NERSC Repositories:
m499, "Center for Edge Physics Simulation: SciDAC-3 Center (C.S. Chang, PI)
mp2, "LLNL MFE Supercomputing" (Bruce Cohen, LLNL, PI)

8.4.1 Project Description

8.4.1.1 Overview and Context

Plasma edge is defined to be the region from the top of the pedestal, about 15% of the outer minor radius inside the magnetic separatrix surface, through the open-field-line scrape-off layer to the material walls. It has been verified experimentally that edge conditions have crucial effect on the core plasma pressure, and thus on the fusion efficiency. Specifically, as the plasma heating power into the core is raised above a threshold value under a clean wall condition --thus, less influx of neutral particles, the edge plasma makes a transition to form a plasma pressure pedestal (see Fig. 1a for a particle density pedestal). This transition is called the L-H transition (low- to high-confinement mode transition). A deep well structure in the radial electric field Er is formed, with a significant reduction in turbulence intensity, within the pedestal layer at the time of the transition. The core ion temperature Ti then rises quickly by a similar amount to the pedestal temperature rise amount. The ITER design requires a Ti pedestal in the range of 4-6 keV to achieve its fusion power mission.

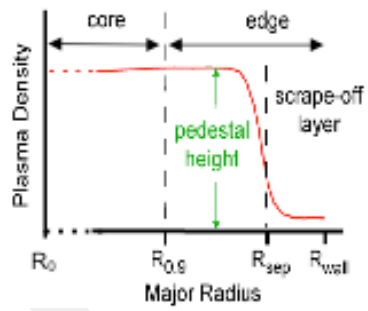


FIG 1a: Plasma edge region

The buildup of sharp pedestal pressure gradients can cause the pedestal to crash from large-scale edge-localized mode instabilities (ELMs). ELMs can rapidly erode the plasma facing material surface via a high-power pulse of particle and energy flux - a serious issue for the ITER program. Few suppression methods against the ELM activities exist. The most promising method found so far is the use of external resonant magnetic perturbation (RMP) coils in an attempt to create stochastic magnetic field structure in the edge pedestal region.

Despite its critical importance and a few decades of research, a predictive understanding of the pedestal physics does not exist. The most severe difficulty has been in that the edge problem is fundamentally kinetic, non-thermal, nonlinear, and multiscale with strong overlapping of scales among various kinetic multi-physics. Extreme scale computing is required to attack the edge problem. It is the purpose of this project to understand such edge physics by performing comprehensive first-principles gyrokinetic simulations using the largest HPC computers, with all the important physics interacting together in multiscale within one code. Physics understanding from kinetic simulation can also improve fluid equations and reduced simulations.

This case study will contain material from the XGC family of kinetic codes that are based on the particle-in-cell (ODE) method, the TEMPEST/COGENT kinetic code suite that is based on the continuum (PDE) method, and the BOUT++ gyro-Landau-fluid code.

8.4.1.2 Scientific Objectives for 2017

XGC Family of Kinetic Particle Codes

The scientific objectives for 2017 in this large-scale edge simulation project is to study all the important edge physics in a single executable XGC1 from the one set of gyrokinetic equations: electromagnetic turbulence, neoclassical physics, neutral particles, impurity particles, wall recycling, wall sputtering, and 3D physics. Simulation will be extended to experimental edge time scale (~50 ms) using a multiscale technique. The electromagnetic XGC1 will have a capability to simulate the MHD-like edge localized mode instabilities (ELMs). However, XGC will also be coupled to BOUT++ and M3D-C1 for MHD/two-fluid simulation of ELMs, in order to approach the ELM problem from two different angles.

Specifically, the physics goals will include 1) L-mode and H-mode edge physics, 2) L-H transition and H-L back-transition physics, 3) pedestal growth and structure physics, 4) edge effect on core plasma confinement, 5) neutral particle and atomic physics, 6) scrape-off layer physics, 7) control of edge localized modes using 3D external perturbation and other means, and 8) heat load distribution on divertor plate.

TEMPEST/COGENT Kinetic Code Suite and BOUT++ Gyro-Landau-Fluid Code

We will conduct the integrated simulations with multi-scale dynamics. The focus will be studies of the slow, self-consistent interaction of magnetic islands with turbulence, and plasma-wall interactions using the state of the art gyrokinetic and gyro-fluid codes for multiple Edge-Localized-Mode (ELM) cycles.

8.4.2 Computational Strategies (now and in 2017)

8.4.2.1 Approach

XGC Family of Kinetic Particle Codes

Our computational challenge is to solve the 5D gyrokinetic equations in realistic diverted tokamak geometry, with the magnetic field structure and plasma profiles imported from the actual experimental data. Since the edge is fundamentally in a non-thermal equilibrium state with all the sources and sinks in action, the conventional perturbation method based on a thermal equilibrium background (the so called "conventional delta-f" method) cannot be used. Our approach in the past was to solve the 5D gyrokinetic equations solely using the particle-in-cell (PIC) method, which has a substantial advantage in utilizing extreme

Our advance approach now is to use a hybrid approach, in which we combine some favorable features of the δf and the continuum methods into our PIC method. We extend the operator-splitting scheme, solve the large-spatial slow-time scale background evolution in 5D continuum space, and use the delta-f PIC method to solve other scale physics –mostly turbulence physics. This approach is found to be highly effective in handling the kinetic electron physics.

Presently, we focus our study to the electrostatic turbulence, aiming for understanding the L-mode physics and the L-H transition physics. The study includes the "ion temperature gradient modes (ITGs)," "trapped electron modes (TEMs)," "resistive ballooning modes," and other low frequency electrostatic drift modes.

In 2017, our computational problem will be extended to include most important edge physics phenomena. Three major extensions shall be in place: 1) electromagnetic turbulence solution for a more complete study of the steep pedestal physics, 2) multiscale simulation for extension of the gyrokinetic study to experimental edge time scale (~50 ms), 3) inclusion of more realistic plasma-wall interaction data. All these advances require strong interaction with the SciDAC math (FASTMath), data management (SDAV),

performance engineering (SUPER), and uncertainty quantification (QUEST) Institutes. In particular, the third extension will be performed in collaboration with the SciDAC Plasma Surface Interactions (PSI) project.

In mathematical structures, the electromagnetic gyrokinetic equations are not too different from the electrostatic equations. Numerically, however, there is a challenge in getting the low odd-parity mode number solutions due to the notorious numerical "cancellation problem" or "ballooning formalism." Thus, past gyrokinetic electromagnetic solutions have been limited to the study of micro tearing modes when it comes to the odd-parity instabilities. Fortunately, there are already on-going efforts in the world gyrokinetic simulation community to resolve this issue, with some published record of success. In this project, our 2017 aim is to have a fully resolved simulation of the electromagnetic turbulence physics including the low-n numbers, in close collaboration with the applied mathematicians and computational scientists.

TEMPEST/COGENT Kinetic Code Suite

The purpose of the TEMPEST/COGENT project is to bring to bear a coordinated effort, utilizing modern computing resources, advanced algorithms, and ongoing theoretical development to progress toward a fundamental, quantitative and predictive understanding of the edge plasma. A key component of TEMPEST/COGENT is the development, verification and validation of kinetic simulations, employing Eulerian methods and associated modern numerical techniques to the solution of the drift- and gyro-kinetic equations, in the complex geometry of the edge plasma, complete with sophisticated collision operators required to accurately treat the broad range of collisionality across the edge region. The Eulerian approach has a strong record of success for core plasma turbulence and transport.

A software infrastructure supports the TEMPEST gyrokinetic edge plasma code. A collection of data structures addresses the specialized parallel distribution and communication functions required to perform mathematical operations on non-simply connected tokamak edge geometries. These structures facilitate the creation of data fields on a union of logically distinct regions (core, scrape-off and private flux) as well as the ability to decompose each region into subdomains for distribution across multiple processors. The data structures are implemented as C++ class objects templated on the configuration or phase space dimension and the data type. Class members include communication functions to perform data exchanges at domain boundaries. These data structures are implemented on top of SAMRAI (Structured Adaptive Mesh Refinement Application Infrastructure) and Chombo (Software for Adaptive Solutions of Partial Differential Equations).

A Python interface provides a scripting front end, allowing the data structures described above to be imported from a module implemented as a shared object library. The data objects can be created and manipulated directly from Python (e.g., for use with a Python-based visualization package), but with all substantial computations and communication tasks performed in compiled code. The data objects can also be passed into C++-compiled physics packages implemented as Python modules, including the GK module that integrates our system of gyrokinetic equations. Because a Python interface also exists for the fluid edge code UEDGE, this approach provides a mechanism for coupling fluid and kinetic edge simulations.

BOUT++ Gyro-Landau-Fluid Code

BOUT++ comes from collaboration between Lawrence Livermore National Laboratory (LLNL), University of York, and a growing number of other institutions around the world. The goal of this project is to develop physics understanding and predictive capability for

edge plasma turbulent transport through a Gyro-Landau Fluid extension of BOUT++ code. This project fills a long-standing gap between the fluid descriptions currently in use (which are intrinsically limited) and full gyrokinetic simulations TEMPEST/COGENT code suite, which are fundamental but a long way from practical utility. This includes developing advanced physics models to capture important kinetic effects in fluid description, and novel numeric techniques to solve a set of fluid moment equations with non-local coefficients in massive parallel computational framework, and validating the simulation models on two largest super-conducting tokamaks (EAST and KSTAR) and the largest US tokamak DIII-D. So-called gyro-Landau fluid (GLF) models can offer significant computational savings (up to 1000 times) by using a moment approach to reduce the phase-space dimensionality from (3d, 2v) or (3d, 3v) to (3d), where "d" and "v" refer to real- and velocity-space dimensions, respectively. These methods use non-local transport coefficients, expressed by integral operators, to describe important kinetic processes such as wave-particle resonances (Landau damping) and the gyration of charged particles around magnetic field lines, and compare favorably to full gyro-kinetic models.

BOUT++ is a modular and flexible fluid plasma code so that the equations to be solved can be easily changed to study various physical effects. This makes BOUT++ a powerful tool that can be benchmarked against existing codes with reduced plasma models, and then extended to understand basic plasma physics in more complete, complicated models. To date BOUT++ has been applied successfully to the study of plasma eruptions from the tokamak plasma edge, showing similarities with experimental results, and to turbulence in the Large Plasma Device (LAPD). Fluid simulations using BOUT++ (and its predecessor, BOUT) have shown good agreement with experimentally observed fluctuation properties in LAPD.

8.4.2.2 Codes and Algorithms

XGC Family of Kinetic Particle Codes

We have three XGC family codes for large-scale simulations of edge physics on NERSC computers: XGC1, XGC0, and XGCa. All three codes include kinetic ions, kinetic electrons, and Monte Carlo neutral particles in realistic diverted tokamak geometry. Guiding center motions of ions and electrons obey common Lagrangian equation of motion in the XGC family codes.

XGC1 is a gyrokinetic code and is the largest of the XGC code family codes. The most recent version XGC1 solves the gyrokinetic Vlasov-Maxwell system of equations in the ODE form using a 3D particle-in-cell algorithm for fast and smaller scale physics, while handling the slow and larger scale physics in the 5D grid space. Mathematically, solving this operator split system is equal to solving the original system in one PIC solver. The fully nonlinear collision operation is then performed using finite difference technique on the velocity grid, with the resulting collision information being mapped back to the particle space. Linear, but conservative, Monte-Carlo collision operator is also included in the code as an option. XGC1 solves for neoclassical, turbulence, neutral and impurity physics all together in realistic diverted tokamak geometry. The sheath at the material wall boundary is described by a "logical sheath" method.

XGC0 is a neoclassical guiding center kinetic code, reduced from XGC1 for a faster evaluation of the background plasma and neutral particle evolution. Even though the particle motion is 3D, the code solves for flux-function electrostatic potential only. Thus, the electrostatic potential is symmetric in both toroidal and poloidal directions.

XGCa is a new axisymmetric gyrokinetic neoclassical code. Unlike XGC0, the electrostatic potential is not symmetric in the poloidal direction and the gyro-averaging operation is

performed to model the steep pedestal at higher fidelity. Since the poloidal electric field may be an important ingredient of the scrape-off physics, XGCa can model the scrape-off region more accurately.

TEMPEST/COGENT continuum kinetic code suite

TEMPEST is an edge gyro-kinetic continuum simulation code for kinetic simulation boundary plasma turbulent transport in a general geometry for magnetic fusion devices. TEMPEST is a full-f gyrokinetic code, to simulate H-mode edge plasmas. This five-dimensional ($\psi,\theta,\zeta,E_0,\mu$) continuum code represents velocity space via a grid in equilibrium energy (E_0) and magnetic moment (μ) variables and 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 so includes boundary conditions for both closed magnetic flux surfaces and open field lines. The same set of gyrokinetic equations 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. The spatial derivatives are discretized with finite differences while a high order finite-volume method is used in velocity space (E_0,μ). A fourth-order upwinding algorithm is used for parallel streaming, and a fifth-order WENO scheme is used for particle cross-field drifts. Boundary conditions at conducting material surfaces are implemented on the plasma side of the sheath. The code includes a range of options for collisions: simple Krook, finite-volume Lorentz, a two-dimensional finite-volume package that can work in second or fourth order and evaluate Rosenbluth potentials either from an assigned background Maxwellian or via callouts to an external Fokker-Planck package that calculates Rosenbluth potentials from the gyrokinetic code's distribution function, or a callout to the Fokker-Planck code to calculate the entire collision operator, with interpolations between the gyrokinetic code's $E_0 - \mu$ variables and the Fokker-Planck code's speed/pitch-angle variables. The gyrokinetic Poisson (GKP) equation is solved self-consistently with the gyrokinetic equations as a differential-algebraic system involving a nonlinear system solve via a Newton–Krylov iteration using a multigrid preconditioned conjugate gradient (PCG) solver for the Poisson equation.

BOUT++ Gyro-Landau-Fluid Code suite

BOUT++ is a C++ framework for 3D plasma fluid simulation in real geometry, including both open and closed field lines, developed in part from the original fluid edge code BOUT. BOUT++ uses the PVODE package, which employs the Newton Krylov BDF implicit method, and a range of finite difference schemes, including 2nd and 4th order central difference, 2nd and 4th order upwinding and 3rd order WENO. Several different algorithms are implemented for Laplacian inversion of vorticity to get potential, such as a tridiagonal solver (Thomas algorithm) and a band-solver (allowing 4th order differencing), and the Parallel Diagonal Dominant (PDD) algorithm by neglecting some cross processor terms. The code has been parallelized with 2D domain decomposition using MPI and with good parallel efficiency up to 10,000 processors. BOUT++ employs an object-oriented approach which separates 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, allowing highly flexible studies of a range of extended fluid, and potentially gyro-fluid, equations.

8.4.3 HPC Resources Used Today

8.4.3.1 Computational Hours

XGC: NERSC was the main source of large scale computing in 2012.

BOUT++: BOUT++ users are around the globe. We used around 4.5 million core hours in 2012 all together.

8.4.3.2 Parallelism

XGC-family kinetic particle codes

We typically use 73,728 Hopper cores to run XGC1, which is approximately half of the total Hopper cores. On the other hand, XGC0 ion runs are typically performed on 1,000 Hopper cores. XGC0 electron physics runs typically use 50,000 Hopper cores. Our XGC family codes could use all the Hopper cores with almost linear efficiency. We use only half of the Hopper capability because of the prohibitively long queue wait time. For multi-petaflop simulation of larger scale XGC1 science, we are forced to use Titan at OLCF.

We try not to run large-scale XGC1 jobs concurrently on Hopper. All XGC1 jobs are independent from each other. However, for smaller size XGC0 jobs that require parameter search, we often submit multiple jobs in the queue. What is important for the XGC codes is the weak scaling in number of nodes. The strong OpenMP scaling applies to the intra-node cores.

BOUT++

In BOUT++, strong scaling experiments are carried out by keeping the total number of grid points in the radial, poloidal and toroidal directions constant. BOUT++ is evaluated in the strong scaling regime, as that is how the developers typically use it. The experiments are conducted on up to 65,536 processor cores. Since the domain decomposition is in the toroidal and poloidal directions (2D), the size of each MPI subdomain becomes smaller with increased concurrency, resulting in more boxes with a fewer number of grid points per MPI box. This results in an increase of the surface to volume ratio (at the highest concurrency of 65,536 there are only two grid points in the poloidal direction). Clearly it may be inefficient to run at this regime, but nonetheless provides an interesting insight into performance trends at high concurrency.

We observe good strong scaling behavior to about 8,192 cores and slight performance degradation at higher levels of concurrency. More detailed analysis shows that BOUT++ performs more calculations than expected for a perfectly strong-scaled experiment. Examining BOUT++ communication overhead shows that computation time scales well to about 16K cores, while communication time begins to plateau at around 4K cores. In fact, the fraction of time spent in communication rapidly increases beyond 2K cores. Interestingly, it saturates at about 20% of the runtime at 16K cores and beyond. As the parallelization scheme reaches its limits, we believe the communication overhead is likely an artifact of the asymptotic limits to the surface to volume ratio.

Due to allocation constraints, we typically use 512 to 1,024 cores for a single job for 4 to 8 hours and then investigate the outputs to see if the results are good. If the results are good, the job would be continued for multiple times and the outputs are investigated each time to determine if the job needs to be continued.

The good strong scale is preferred as it shorten the wall clock time by increasing the number of cores, so developers can quickly get the computational results without long waiting period of time.

XGC family kinetic codes

For scratch we currently use about 20 TB/project. One person uses 7TB of scratch space.

We use about 1 TB/project for NGF and have about 300TB stored on HPSS.

We have a large amount of turbulence data to store and analyze. The analysis data on scratch space can be up to 7TB at a time for one designated user.

Scratch space is used for check pointing, data outputting, as well as job submissions. Some times, multiple jobs are submitted for different physics study, multiplying the size of data in scratch space. The amount of data from source and input files in the scratch space is negligible compared to other files.

Permanent storage space is used mainly for storage of the source files, input files, analysis scripts, important output data that needs to be analyzed and accessed often, and important analysis results such as figures and movies.

HPSS is used for storage of complete file directories for all the meaningful runs. This includes all the source codes, all the input files, all the checkpoint data, all the output files, and all the analysis and movie results.

We need to share data between Hopper and Carver. Most of the large-scale jobs are executed on Hopper, but most of the large-scale data analyses are performed on Carver. . We used Carver for data analysis mainly because of the larger memory capability for MATLAB analysis. For large-scale runs, we normally need to load up a pretty large size data in the node for the MATLAB analysis. With the recent installation of MATLAB on Hopper and Edison, we no longer require Carver for this analysis.

We use ADIOS. Other traditional methods slow down our large-scale runs. Periodic update of Adios libraries on all the future large-scale machines will be helpful.

XGC-family kinetic particle codes

Our current estimate is that we will need 2B Hopper equivalent hours for CY 2017 to reach all the scientific goals listed in Sec. 1.2 at high fidelity. We expect to receive some allocations from Titan, mainly for jobs requiring 100-PF range computing. The primary factor for driving the need for this amount of computing time is the electromagnetic electron turbulence physics for realistic edge time scale (~50 ms), using the real electron mass in 3D magnetic perturbations. Real mass electrons require 60 times smaller time step than deuteron. Currently, we use electron mass of 100 times lighter than deuteron, thus requiring only 10 times smaller time step than deuteron. Also, current simulation is for a few ms physics time scale only that is adequate for turbulence saturation.

TEMPEST/COGENT continuum kinetic and BOUT++ gyro-Landau-Fluid code suites

The 5D codes are computationally very demanding and require up to ~512´128´256»10^7 grid points in configuration space and 30´30 in their 2-D velocity space. TEMPEST typically

requires 2048 processors for a single predictive ITER simulation. The time per TEMPEST simulation should be about 24 hours (a few times greater than a traditional fixed-gradient run), implying a total predictive simulation time of 24 × 2048 = 49,152 CPU-hours for a steady state turbulence.

BOUT++ typically uses over 4,096 processors for a single simulation leading to filament formation and eruption for 96 hours with a resolution 516x256x128 in single-null divertor geometry for a single ELM crash.

We plan to conduct integrated simulations with multi-scale dynamics with multiple ELM cycles.

Our plan will also leverage international resources and international tokamak data for validation of the models to investigate the characteristics of ELM deposition patterns for long-pulse discharges.

Therefore, these two code suites will add up to a total computing resource allocation is 50,000,000 CPU-hours in 2017.

8.4.3.3 Parallelism

XGC-family kinetic particle codes

We expect to use as many Hopper equivalent cores as are available in 2017, assuming the 2017 system has approximately the same number of nodes as existing systems. Our codes are proven to scale efficiently to the maximal number of Jaguar nodes and cores. We do not anticipate running more than one large job concurrently.

8.4.3.4 Data and I/O

XGC-family kinetic particle codes

Scratch space: 600 TB. 32X higher capability means roughly 32X more scratch space, mainly due to the proportional increase of the checkpoint data in our PIC code XGC1.

Permanent: 10TB. The output data size and the analysis results will not grow as fast as the computing capability or checkpoint data. 10X increase between now and 2017 is estimated to be 10TB.

HPSS: 3 PB. Since many of the non-essential checkpoint files can be discarded, we anticipate that 10X more HPSS is required from 32X higher computing capability.

8.4.3.5 Memory

XGC-family kinetic particle codes

Assuming that a 32X NERSC machine in 2017 will have 10,000 nodes, the minimal memory requirement of our XGC code can be anticipated to be 50 GB per node with 500 TB of aggregate memory. A large-scale, full-function PIC code is fundamentally a low memory-per-node code.

8.4.3.6 Many-Core and/or GPU Architectures

XGC-family kinetic particle codes

XGC1 has already been ported to heterogeneous GPU-CPU computers (Todi in Switzerland and Titan at OLCF). Unlike in other GPU codes, XGC uses CUDA-Fortran. Our codes will be adapted to any other futuristic type of "lightweight" cores and accelerators in collaboration with the SciDAC SUPER Institute, if necessary.

8.4.3.7 Software Applications and Tools

XGC-family kinetic particle codes

This is not an easy question to answer since new HPC software will be developed quickly. We will need Adios, PETSc, GPU performance analysis tools, and the compilers for FORTRAN codes. We anticipate continuing the MATLAB library usage in 2017.

8.4.3.8 HPC Services

We will require consulting and account support, training, web interfaces and others.

8.4.3.9 Time to Solution and Throughput

XGC-family kinetic particle codes

Our aim is to get the answer within 24 wall-clock hours of simulation from our large jobs. Larger scale problems will be submitted to proportionally larger number of nodes. A queue policy to favor large-scale jobs will be important.

8.4.3.10 What Else?

XGC-family kinetic particle codes

Improving the inter-node communication speed will be the most important HPC feature.

8.4.3.11 Requirements Summary

XGC1 Kinetic Particle Code Summary	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (million)	78	2,200
Scratch storage and bandwidth	20 TB	600TB
	35GB/sec	350GB/sec
Shared global storage and bandwidth (/project)	0.5 TB	10TB
	15GB/sec	150GB/sec
Archival storage and bandwidth (HPSS)	15 TB	3,000 TB
	1GB/sec	10GB/sec
Number of cores* used for production runs	73,728	3,000,000
Memory per node	5GB	Minimum of 50GB
Aggregate memory	30TB	Minimum of 500TB

Draft

8.5 Study of the Internal Dynamics of ITER

Principal Investigators: Stephen C. Jardin (PPPL), Carl Sovinec (University of Wisconsin), Linda Sugiyama (MIT)

NERSC Repositories Represented: mp288, m876, mp21, mp200, m455, m908

8.5.1 Project Description

8.5.1.1 Overview and Context

The ITER device, now being constructed in Cadarache, France, is a monumental step towards the realization of fusion power. The $20B experiment, being financed by 7 international governments including the U.S., is the same scale as a follow-on fusion power plant. Success in this endeavor will be a turning point in the quest for a sustainable, carbon free, and safe source of energy for the planet.

There are several aspects of the control and operation of a magnetically confined fusion device like ITER that can benefit from high-end computation. We are concerned with studying the internal global dynamics of the high-temperature ionized gas (plasma) in ITER, primarily with two world-class extended magnetohydrodynamics (MHD) codes, NIMROD and M3D-C^1, that have been developed by us through our SciDAC grants over the last decade, but also using newer exploratory techniques involving Finite Time Lyapunov Exponents (FTLE) that will be further developed during this time period. We are presently making extensive use of NIMROD and M3D-C^1 to perform similar studies in existing fusion devices for further verification and validation.

A comprehensive and reliable simulation capability for the MHD phenomena that will occur in ITER will be invaluable in a number of ways. It will provide insights into the different modes of operation that are possible with the device. It will also allow us to test optimization and control strategies. It will certainly give the U.S. a competitive edge over our international partners in developing proposals for run time once ITER becomes operational.

8.5.1.2 Scientific Objectives for 2017

The primary research objective is to extend the studies performed on existing fusion devices by using plasma parameters (i.e. temperatures and magnetic fields) much higher and hence closer to those expected in ITER. Higher temperature implies lower plasma resistivity, which implies that the magnetic Lundquist number S will be higher. This, in turn, necessitates a finer mesh as there are localized reconnection layers that need to be resolved that scale in size as $S^{-\alpha}$ with $\alpha \sim 1/3 - 1/2$. It will also increase the importance of two-fluid (separate electron and ion dynamics) and kinetic effects that are outside the scope of the basic resistive-MHD model. The reconnection layers that will develop in these studies are localized around a few rational surfaces or at the plasma edge and so the adaptive meshing capability of the codes will be very useful in distributing the mesh points where they will be most effective. Present nonlinear studies performed with these codes used high-order finite elements with typical (normalized) element dimension of 0.01 with 32-64 toroidal planes (or toroidal Fourier modes) and this restricted us to values of $S=10^6$. Through a combination of better mesh adaptation and more processors, we plan to have an effective element size of 0.0025 or less at the reconnection layers and to include 128-256 planes or toroidal Fourier modes. This should allow computations with $S=10^7$ or 10^8 or higher, which is well within the range of existing tokamaks. This will allow detailed

validation of our results and give us increased confidence to scaling these results to ITER parameters.

We now have close collaborations with several experimental teams that are performing detailed diagnostic measurements of internal reconnection events (sawtooth), edge localized modes (ELMs), and tearing-mode phenomena in the DIII-D and KSTAR tokamaks. A new 2D electron cyclotron imaging diagnostic on both these machines allows one to obtain unprecedented time dependent images of the evolution of the electron temperature during an internal reconnection event or ELM crash, or the growth of a magnetic island. For example, the diagnostic measurements show that there can be a wide variety of types sawtooth behavior, with crash times ranging from 100 μs to 10 ms in KSTAR, depending on plasma parameters and modes of operation. We also see a wide range of sawtooth behavior in our simulations, although the simulations are presently at lower values of plasma temperature than the experiments and so cannot directly be compared. With the expected growth of computer resources in the next few years, it should be possible to run a number of simulation cases with realistic parameters (temperature and Lundquist number) so that results can directly be compared with these experiments.

We expect this close interaction between new state-of-the art diagnostics with new state-of-the-art simulations to enable a much deeper understanding of the stability and magnetic reconnections processes occurring in high-temperature tokamak plasma. This understanding and simulation capability can directly be applied to ITER. Extensions of this work in future years will be aimed at simulating several sawtooth, ELM, and tearing mode control techniques that have been demonstrated experimentally and projecting these to ITER.

8.5.2 Computational Strategies

8.5.2.1 Approach

To put our studies into context, we note that there are 4 classes of simulation codes used by fusion researchers to simulate and study phenomena in magnetic confinement devices: (a) The RF codes such as TORIC and AORSA address the interaction of externally imposed electromagnetic radiation with the plasma as a heating method; (b) The micro-turbulence codes (or gyrokinetic codes) such as GYRO, GTS, and XGC1 study the fine-scale turbulence that leads to increased heat and particle transport; (c) The extended MHD codes being used in the present study, NIMROD and M3D-C^1 and exploratory projects such as FTLE, which study the nonlinear evolution of global instabilities and magnetic reconnection events; and (d) The transport codes which study the evolution of an assumed stable discharge over very long timescales, assuming the transport coefficients are known.

The NIMROD and M3D-C^1 simulation codes, addressing class (c) phenomena in the above, are capable of solving the nonlinear extended MHD equations for initial-value computations. Despite stiffness associated with multiple time-scales, the codes efficiently advance the physical fields in three spatial dimensions using implicit algorithms that can take very large time steps compared to the Courant condition based on the MHD wave velocities. Once the appropriate transport coefficients, boundary conditions, and sources and sinks are specified, we can compute the slow (transport time scale) evolution of the pressures, densities, and current density and determine when the system evolves into an unstable state. The codes can then calculate the nonlinear consequence of that instability and study proposed control techniques.

8.5.2.2 Codes and Algorithms

The NIMROD and M3D-C[1] codes are similar in many respects but differ in many implementation details. Both codes solve the extended MHD equations using a split-implicit algorithm and high-order finite elements. The M3D-C[1] code uses finite elements in all three dimensions whereas the NIMROD code uses a Fourier expansion in the toroidal direction. The codes differ in the internal representation of the vector fields representing the velocity and the magnetic field and in the way that the divergence constraint on the magnetic field is implemented.

With both codes, each time-step involves several computation- and communication-intensive operations. Projections of the time-differenced equations onto finite-element test/basis functions are computed with numerical spatial integration and partially gathered into algebraic vectors and matrices. With its 1D Fourier expansion, NIMROD also performs FFTs as part of this step using all-to-all communication within processor groups formed by 2D decomposition of the perpendicular plane. The algebraic systems are solved with preconditioned Krylov-space methods. Both codes use block-based preconditioning with large diagonal blocks solved by direct methods for sparse matrices (typically SuperLU_DIST). This general approach is dictated by the ill conditioning of the matrices, which results from the stiffness of the problems. M3D-C1 and NIMROD differ somewhat in terms of the decomposition that defines the blocks, and M3D-C1 has real algebraic systems, whereas NIMROD has complex systems.

8.5.3 HPC Resources Used Today

8.5.3.1 Computational Hours

Our primary computational resource is NERSC. In 2012 repo mp288 used 7.9 million hours at NERSC and repo mp200 used 1.4 million hours. The total for this case study was 9.3 million.

8.5.3.2 Parallelism

M3D-*C*[1]: Typical runs today use 1,000-3,000 cores, but we have demonstrated in weak scaling studies that the algorithm scales acceptably to over 12,000 processors for sufficiently large numbers of finite elements (or mesh points). We do not have enough computer time allocation to routinely use the finest mesh is the reason that we use fewer cores (and a coarser mesh) than the maximum. Our group typically has 4-5 projects in the job queues at one time, each of which requires 5-10 restarts to perform a complete simulation. Weak scaling is important to us, as we want to run with higher resolution using more processors.

NIMROD: Most production computations with NIMROD use several hundred to one or two thousand cores. NIMROD has also demonstrated weak scaling to more than 10,000 cores using MPI alone and to more than 50,000 cores using MPI combined with OpenMP. However, before investing a significant fraction of resources in a large computation, we find it essential to perform many smaller computations to understand linear spectra, to assess the potential significance of different physics contributions, and to estimate resolution requirements. Having multiple jobs running concurrently is common, but the computations often represent different applications, so they may be of different size or duration. Like computations with M3D-*C*[1], weak scaling is most important for achieving realistic parameters, but improved strong scaling would also be welcome for better throughput of small to medium-sized computations. At present, generating data that comprise our algebraic systems scales well in a weak-scaling sense and in a strong-scaling sense (to a

reasonable degree). Solving the algebraic systems with parallel scaling is more challenging. The matrices differ significantly from those generated by standard parabolic and elliptical PDEs, and efforts to apply multi-grid methods have not shown clear benefits. The sparse direct solves for our block-based preconditioning are communication-intensive, and parallel solution of the algebraic systems is the most limiting aspect of the algorithm.

8.5.3.3 Data and I/O

M3D-C1 I/O: The initial input to this code is minimal. Normally we need only a short namelist, a grid definition file, and optionally a small file of a standard type that specifies the 2D initial conditions (equilibrium file). In the interest of scalability, we have one process that reads and broadcasts the input. The output files are of two types: restart and graphics. The restart file (or checkpoint file) is written using ADIOS and contains all the information required to restart. The graphics files are hdf5. There is a separate hdf5 file with all the DOF at each specified time point, but also a single hdf5 file with all the time-dependent scalar information (such as the kinetic energy in each toroidal mode). The output accounts for only a small fraction of the run time. We have several post-processing programs that can read these hdf5 files for processing, including VisIt.

NIMROD I/O: A "dump" (checkpoint) file containing the entire system state at a given time is read at startup and written periodically, usually with hundreds of time-steps between these I/O operations. In the past, the requirements for NIMROD have also been modest, with each dump file being smaller than 100 MB. The data is communicated to a single processor that performs a binary write operation. In recent large computations, the files can exceed 0.5 GB. We have experience with hdf5 files and intend to convert to parallel read and write operations in this format when I/O becomes a performance concern.

M3D-C^1 Data Storage: The data that is stored is in the form of the hdf5 files. Each time slice now produces about 1 GB of data. We typically keep 100 time slices per run, which would be about 0.1 TB per run. However, in some time periods we need more frequent output, so that 0.2-0.5 TB per run would be a safe estimate. We perform on the order of 10 major runs per year, so this would be an upper bound of about 5 TB of data that needs to be stored each year. We typically keep this in the project space for the order of 1 month and then move it to HPSS. We do not use extra scratch disk space except to store the output files during the run or to store the restart files between run segments. We do not normally share data between NERSC systems.

NIMROD Data Storage: The permanent storage requirements for NIMROD are similar to those for M3D-C^1, approximately 5 TB per year. The present temporary-space and user home space are adequate. Having a number of code users who are not developers, the NIMROD project takes advantage of /project space and sharing between systems. It facilitates collaboration and helps users access executables and data. The size of our present /project space is on the order of 1 TB, and needs will likely increase by approximately 0.5 TB per year. Data sharing is primarily between the large supercomputers, Hopper, Carver, and now Edison.

8.5.4 HPC Requirements in 2017

8.5.4.1 Computational Hours Needed

M3D-C^1: We need to increase our spatial resolution by a factor of 4 in each of 2 dimensions and a factor of 2 in the third dimension which implies at least 32 times more compute power assuming our implicit algorithms exhibit perfect weak scaling. Scaling from 2012

usage, this would be in the order of 100-200 Million CPU hours. In addition to the increase in compute time coming from using more spatial resolution, we are increasing the fidelity of our physics model (with more two-fluid and kinetic effects), which also results in increased run time for the same zone size. Some of this time may come from successful INCITE proposals, but we have no guarantee of this. Editor's Note: the value of 200M in the table below represents scaling to the entire repository based on AY 2012 usage.

NIMROD: Increased spatial resolution is also a priority for NIMROD computations. Another priority is applying more realistic modeling. Computations with the two-fluid model address plasma drift and fast-reconnection effects that are not considered by resistive-MHD. Per time-step, the computational cost increases by a factor of 2-3 relative to resistive-MHD, resulting from the magnetic-field advance having a more ill conditioned matrix. Drift effects in this model can lead to the presence of faster oscillations that need to be resolved temporally. This will force us to use greater temporal resolution, possibly by as much as a factor of 10. Thus, relative to our present allocation of approximately 4 million CPU hours per year (divided among 4 repositories), our needs will be as large as 100 million hours per year, assuming successful scaling on new hardware.

8.5.4.2 Data and I/O

M3D-*C*[1] and NIMROD: All of our storage requirements scale linearly with the number of mesh points (or finite elements) and so we expect these to increase by a factor of 32. We are not presently I/O limited, and so do not expect to require increased I/O rates.

8.5.4.3 Scientific Achievements with 32X Current Resources

M3D-*C*[1] and NIMROD: The numbers listed above are 32 times our present resources.

8.5.4.4 Parallelism

M3D-*C*[1]: Our goal is to use between 20,000 and 100,000 processors per job in 2017. **NIMROD:** By using threading together with MPI, we expect to run large computations on 100,000 cores by 2017.

For all codes: We expect to have several simultaneous submissions in the job queues at any one time, but each submission would be only a single job. Many more computations at the 1000-core scale will still be required, however. In some cases, it may be efficient to bundle smaller computations into a single job, but we do not expect this to be standard procedure.

8.5.4.5 Memory

M3D-*C*[1]: Our jobs now on Hopper are typically memory limited. For a given number of degrees of freedom (DOF) (which is the number of finite elements times the DOF per element), we have to keep increasing the number of nodes on Hopper until we have sufficient memory to run the job. Having more memory per core for a given number of DOF would allow us to use fewer cores and achieve better scaling. The memory requirement scales slightly worse than the total number of DOF because we use a block-Jacobi preconditioner based on SuperLU_dist which has a large fill-in factor. We think that a memory/core ratio of about 4GB (such as in Edison) would be close to optimal in the next 3-4 years.

NIMROD: Memory also limits resolution in large NIMROD computations. The recent work on threading and development to obtain sparsity patterns for SuperLU_DIST without duplicating integer-array storage have been motivated by memory considerations. Each of these efforts represents significant progress, but greater memory per core would increase

flexibility. On a shared-memory workstation, for example, we benefit by having more than 4 GB per core.

8.5.4.6 Many-Core and/or GPU Architectures

M3D-*C*[1]*:* Our implicit time-step has 3 major parts to it: (1) define the matrix elements by performing 3D integrations, (2) apply the pre-conditioners, and (3) solve the linear system. We think that part (1) will benefit greatly from GPUs and have started exploratory studies. For parts (2) and (3), we use the PETSc SuperLU_dist based block Jacobi preconditioner and GMRES Krylov solver, so that we would need developments by the PETSc group to take advantage of GPUs in these solver areas.

NIMROD: Our efforts at threading address the generation of information for algebraic systems, which does not require global communication. Other parts of the algorithm may also benefit, but solving algebraic systems will be the bottleneck. New approaches and development from the computer science community are welcome. However, it is worth emphasizing that global propagation of information is associated with problem stiffness from the application and requires frequent movement of data. (Here, we are distinguishing "global" from the MPI "collective" definition. There are collective communication operations among distinct communicator groups when generating algebraic systems, but they do not cover the entire domain. Conversely, solving algebraic systems effects global communication without collective operations.)

8.5.4.7 Software Applications and Tools

M3D-*C*[1] and NIMROD: HDF5, ADIOS, PETSc, VisIt, Fortran-90 compilers
SuperLU_DIST, OpenMP

Many of our users strongly prefer AVS-Express to VisIT for visualization.

8.5.4.8 HPC Services

M3D-*C*[1] and **NIMROD:** NERSC excels in this area, and the NIMROD project would benefit from tutoring and assistance with new hardware architecture.

8.5.4.9 Time to Solution and Throughput

M3D-*C*[1], NIMROD and FTLE: Given the scientific and computational importance of jobs at the hundreds to 1000-processor level, increased throughput for this scale would be a tremendous benefit. Also, many of our jobs require long wall-clock times at moderate size.

8.5.4.10 Data Intensive Needs

M3D-*C*[1] and NIMROD*:* No change from today.

8.5.4.11 Requirements Summary

Repo mp288, representing M3D-*C*[1] usage	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (Hopper core-hour equivalent)	7.9 Million	200 Million
Scratch storage	0.1 TB	3 TB
Maximum number of cores* that can be used for production runs		
Shared global storage (/project)	5.6 TB	16 TB
Checkpoint bandwidth		
Archival storage and bandwidth (HPSS)	3.5 TB	160 TB
Maximum I/O bandwidth (excluding checkpoint data)		
Number of cores* used for production runs	700- 3,000	20,000-100,000
Memory per node	1.5 GB/core	4 GB/core
Aggregate memory	4.5 TB	400 TB
Computational Hours (Hopper core-hour equivalent)	4 Million	100 Million
Repo mp200, representing NIMROD usage	**Used at NERSC in 2012**	**Needed at NERSC in 2017**
Computational Hours (Hopper core-hour equivalent)	1.4 Million	100 Million
Scratch storage	0.1 TB	2 TB
Shared global storage (/project)	0 TB	4 TB
Archival storage (HPSS)	27 TB	200 TB
Number of cores* used for production runs	100-2000	1,000-100,000
Memory per node	1 GB/core	4 GB/core
Aggregate memory	1 TB	400 TB

Draft

8.6 Small Scale Plasma Experiments: Simulations and Model Validation

Principal Investigators: Vyacheslav S. Lukin, (Naval Research Laboratory), Brian A. Nelson (University of Washington), Carl R. Sovinec (University of Wisconsin).

NERSC Repositories: m489, m1255, mp200, m172

8.6.1 Project Description

8.6.1.1 Overview and Context

Small-scale plasma experiments play an important and highly valuable role in the DoE Fusion Energy Sciences research portfolio. Primarily, though not exclusively, focused on investigating different aspects of magnetic plasma confinement with the ultimate goal of constructing a reliable fusion energy reactor, small-scale laboratory experiments often serve as test-beds for new ideas towards improving the design of a future reactor. An equally important role of these experiments is their ability to study in isolation limited subsets of plasma phenomena that are known to be of concern, and have to be better understood and modeled, for the much more powerful and complex ITER-scale plasma experiments to operate reliably.

Over the past decade, with increased availability of computing resources and software, as well as the greater recognition of the intrinsic value of numerical modeling for understanding the behavior of highly nonlinear systems, advanced simulations have become an integral part of the research process on many of the small scale plasma experiments. Magnetohydrodynamics (MHD) based, particle based, and/or hybrid numerical plasma models developed by the greater computational plasma physics community are often employed for both idealized and whole-device simulations of these experiments. The Plasma Science and Innovation Center (PSI-Center), a multi-institution collaboration headquartered at the University of Washington, has been serving as the conduit to enable and support such modeling efforts for many of the small-scale plasma experiments.

The two workhorses of the PSI-Center are NIMROD, an extended MHD code with hybrid capability, and HiFi, a multi-fluid modeling framework for solving complex systems of coupled partial differential equations (PDEs). The PSI-Center also employs, develops and improves a variety of other numerical codes, such as the 3D force-free plasma equilibrium solver on an unstructured tetrahedral mesh PSI-Tet, the DCON MHD stability code for axisymmetric toroidal plasmas, and the particle trajectory-tracing code PUSH.

8.6.1.2 Scientific Objectives for 2017

Over the next several years, the mission of the PSI-Center will broaden from being primarily a support center for simulations of small-scale plasma experiments, to proactively performing systematic plasma model validation studies using the experimental data. It is for such validation studies that the ability of the smaller laboratory experiments to focus on a limited subset of all the plasma phenomena observed in larger and hotter machines will become critically important.

The ability to conduct relatively simple, well-diagnosed plasma experiments with high repetition rate at low cost allows for better quantification of the sources of uncertainty both in the experimental data and in the corresponding simulations of the small scale experiments. Taken together with the more detailed and more accurate simulations that

will be enabled in the next four years by the expected improvements in the high performance computing hardware and software, the PSI-Center's goal for 2017 is to be able to quantitatively estimate the validity of the plasma models implemented in the NIMROD code and the HiFi framework in the parameter regimes of these experiments.

Doing so will require substantially increased computational time allocations and much greater throughput that allows to perform many medium-sized simulations that complement the "one-off" biggest possible simulations by exploring the available parameter space and performing statistical studies around the performance point of any given experiment.

8.6.2 Computational Strategies

8.6.2.1 Approach

By far the most computationally intensive calculations presently performed and supported by the PSI-Center are two- and three-dimensional initial value extended or multi-fluid MHD calculation performed using the HiFi framework and the NIMROD code. (For the purposes of this case study, the discussion of the PSI-Center's current usage and future needs will be limited to HiFi and NIMROD from here on forward.) A typical calculation is either driven to or is initialized in an unstable state, which in turn drives the nonlinear plasma dynamics of the experiment being simulated; an example of a HiFi simulation of the Swarthmore Spheromak Experiment is shown in Figure 1. If the experimental conditions allow for discharges with sufficiently long pulse length, multiple dynamical cycles may need to be simulated.

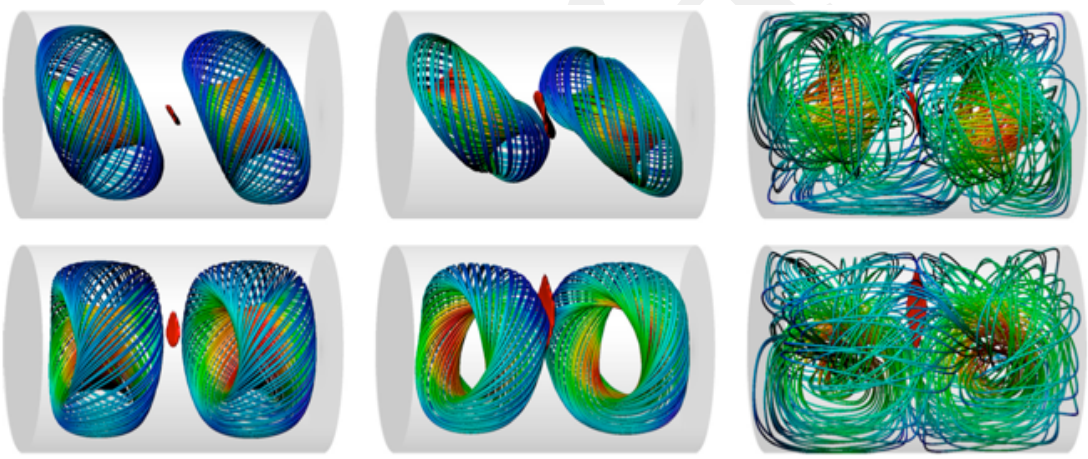


Figure 1. HiFi simulation of reconnection and merging of two spheromaks in the Swarthmore Spheromak Experiment. Representative magnetic field lines illustrating the two spheromaks, are shown here. Peak in current density is represented by the surface in the center of the volume. The top row and bottom row both show the same point in time; the images in the bottom row are an orthogonal view to the ones in the top row. [from Gray et al., Physics of Plasmas 17, 102106 (2010)]

8.6.2.2 Codes and Algorithms

Both NIMROD and HiFi are implicit initial value parallel codes that employ semi-structured meshes and use up to thousands of computing cores for two- and three-dimensional simulations of the PSI-Center supported experiment. However, while similar in many ways, there are several important differences between the algorithms and the implementation approaches used in the two codes.

Both codes use semi-structured arbitrary order finite (spectral) elements spatial discretization in a two-dimensional plane; but while NIMROD uses the Fourier representation in the third dimension, HiFi treats all three dimensions on equal footing. For grid generation in complicated experimental geometries, HiFi relies on the semi-structured grid generator package CUBIT, while NIMROD uses a homegrown grid-generation algorithm. HiFi framework's temporal advance is fully implicit i.e., the full system of PDEs is always advanced in a single implicit time-step, which together with the generic user interface gives HiFi the flexibility of a general PDE solver. On the other hand, NIMROD uses a split-implicit time advance algorithm with hard-wired PDEs, which allows it to have a smaller memory footprint. Both codes use preconditioned Krylov-space methods for solving the resulting large sparse algebraic systems: the NIMROD solver is customized with an interface to SuperLU_DIST, and HiFi relies primarily on the PETSc interface for access to a variety of preconditioning and linear solver libraries.

8.6.3 HPC Resources Used Today

8.6.3.1 Computational Hours

The PSI-Center and the collaborating small scale experimental programs included in this case study, namely SSX, HIT-SI and Pegasus, all rely on the NERSC computational resources to a large extent. However, local small (up to ~200 core) computing clusters also play a very important role in the day-to-day research. Multiple surveyed scientists reported running local 25 to 100 core jobs, both in the code testing and the production simulation phases, on a daily basis.

8.6.3.2 Parallelism

The size of PSI-Center production runs using both HiFi and NIMROD can vary by about two orders of magnitude, from using only 50 or so and up to about 5,000 cores, depending on the plasma parameter regime of the particular experiment being modeled, whether the simulation is done in two or three dimensions, and how detailed is the physical model used in the simulation. The codes have demonstrated strong scaling up to a couple hundred cores, and, for some applications, weak scaling up to $O(10^4)$ cores. The biggest one-time simulations have been performed on over 50,000 cores.

There are two primary reasons for typically using fewer cores than one could:

(1) CPU hours is a limited resource and therefore we want to use them efficiently. Thus, due to lack of strong scaling beyond a couple of hundred cores, if certain amount of spatial resolution is sufficient and the problem fits on fewer cores and can run in reasonable time, using fewer cores is advantageous.
(2) Much of the work we do is focused on understanding the underlying nonlinear physical processes that are at play in the experiments we model. The typical process for doing so usually involves many iterations with sensitivity studies to both numerical and physical model-dependent parameter variations. When for these purposes a smaller problem size is acceptable, due to the lack of strong scaling and limited available resources, using fewer cores is advantageous.

To perform the parameter scans, any given user running the codes may have up to 3 jobs in the queue at any one time. Some jobs may require one or two restart due to the NERSC job runtime limits.

8.6.3.3 Data and I/O

Scratch (temporary) space: We use about 700 GB for storing running simulation checkpoint files, synthetic data analysis, and visualization.

Permanent (can be shared, NERSC Global File System /project): We use about ~2-5 GB for storing libraries, source code and executables.

HPSS permanent archival storage: We use about 2-5TB for data backup, and archiving past calculations for long term storage

Data are shared between Hopper and Carver for running simulations, and Euclid for post-processing and visualization with VisIt.

In NIMROD, I/O is performed in a serial fashion with a single processor responsible for reading (writing) binary checkpoint files and appropriately distributing (collecting) the information to (from) the rest of the computing cores. In HiFi, both serial and parallel hdf5, as well as the serial binary format, are available as run-time I/O options for reading and writing checkpoint files.

8.6.4 HPC Requirements in 2017

8.6.4.1 Computational Hours Needed

Given the uncertainty in the future domestic DoE Fusion Energy Sciences budget as a whole, and the funding for the small-scale plasma experiments, in particular, it is difficult to estimate the computational hours that may be needed by PSI-Center and the collaborating experiments in 2017. With that said, and assuming a sufficient number of research scientists and graduate students is available to do the work, the expected need will be in the range of 100-200 Million CPU hours. The primary reason for the increased need will be driven by the combination of two factors:

(1) The change in the mission of the PSI-Center towards model validation, as described above. The broader scope will require running a substantially greater number of simulations than we do now around the performance point of any particular experiment in order to quantify the uncertainty associated with the spatial resolution, initial and boundary conditions used in the simulations. Similarly, the sensitivity of the results to the specific physics model and the adjustable parameters used in the simulations will need to be evaluated. This alone will account for up to 100-fold increase in the demand for computing time.
(2) The increased maturity and better performance of the codes will allow the PSI-Center's personnel to devote more time to running simulations of the experiments and validating the models, rather than developing the software. Improvements in the codes' user interface will enable more users directly affiliated with the experiments and without significant computational experience to run the codes themselves. Together, these may account for another factor of two to three increase in the demand for NERSC compute hours.

8.6.4.2 Parallelism

Continued improvements in the scalability of the software specific to HiFi and NIMROD, as well as improvements in the external libraries the codes rely upon, are expected to yield a factor of two to four improvement in the overall scalability of the codes. This will lead to

users running more simulations in the 10,000 to 20,000, and up to 100,000, cores per job range.

The increased availability of computing time and better throughput for 100-core to 1,000-core scale jobs will also result in many more users running those jobs instead of the 50-core jobs they run now.

The expected increase in the user base and the number of simulations of small scale plasma experiments necessary for model validation is likely to yield O(10) jobs of varying size in the queue at any one time.

8.6.4.3 Data and I/O

The permanent storage capacity is expected to scale linearly with the number of users, thus increasing by about a factor of two.

The scratch space capacity is expected to scale linearly with the number of simulations being performed simultaneously and the size of the computational grids, thus likely increasing by a factor of O(10).

The HPSS archival storage is expected to increase linearly with the number of simulations performed over the course of a year, thus likely increasing by a factor of O(100).

No I/O bandwidth limitations for either HiFi or NIMROD have been identified so far.

8.6.4.4 Scientific Achievements with 32X Current Resources

The expected demand for CY 2017 described above already exceeds what would be available with the annual doubling of allocated compute hours.

8.6.4.5 Memory

The minimum memory requirement that maintains the parallel scaling's quoted above is 2 GB per core. A more optimal memory per core ratio would be 4 GB per core.

8.6.4.6 Many-Core and/or GPU Architectures

At this time, both NIMROD and HiFi rely on MPI communication and run on homogeneous CPU systems. Initial development and planning for using heterogeneous many-core architectures has began, but any help from the computer science community and NERSC will be welcome. It should be noted that a GPU-capable version of the PETSc library is now available, however GPU-enabled versions of such linear solvers as SuperLU_DIST and MUMPS, or similar alternatives, would have to become available for either code to be able to take full advantage of the new heterogeneous architectures.

8.6.4.7 Software Applications and Tools

PETSc, HDF5, NetCDF, SuperLU_DIST, MUMPS, HYPER, VisIt, Fortran-90 compilers

8.6.4.8 HPC Services

Maintaining high standards in account support, timely installation of new software versions, as well as continued training and tutoring in the use of new systems is all that is necessary.

8.6.4.9 Time to Solution and Throughput

As indicated above, increased throughput for 100 to 5000 core size jobs, possibly also allowing for longer runtimes for a single job, would be very helpful.

8.6.4.10 Data Intensive Needs

None.

8.6.4.11 Requirements Summary

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours (Hopper core-hour equivalent)	3.1 M	200 M
Scratch storage	1 TB	10 TB
Shared global storage (/project)	2.6 TB	26 TB
Archival storage (HPSS)	5 TB	200 TB
Number of cores* used for production runs	50-5000	200-100,000
Memory per core	1-2 GB/core	2-4 GB/core
Aggregate memory	10 TB	400 TB

Draft

9 General Plasma Science Case Studies

9.1 Petascale Kinetic Simulations in Laboratory and Space Plasmas

Principal Investigator: Homa Karimabadi (UCSD/SciberQuest, Inc.)
Worksheet Author: Homa Karimabadi (UCSD), Kai Germaschewski (UNH)
Other contributors: Y. Omelchenko (UCSD/SciberQuest, Inc.), William Daughton (LANL), Vadim Roytershteyn (SciberQuest, Inc.), Mahidhar Tatineni (SDSC), Amitava Majumdar (SDSC), Burlen Loring (LLBL).
NERSC Repositories: m148, m1303

9.1.1 Project Description

9.1.1.1 Overview and Context

Our research concerns the three fundamental areas in plasma physics with direct relevance to astrophysical and laboratory plasmas: a) magnetic reconnection, b) turbulence, and c) magnetic field generation due to the dynamo process. The proper description of these processes often requires treatment of multi-scales ranging from micro (electron) to macro (ion) to mesoscale and poses a major computational challenge. The reason is that the plasma environment in many of the applications of interest such as planetary magnetospheres or the solar wind is in the collisionless regime, where electron and ion kinetic effects play a critical role. In addition there exists a large class of astrophysical environments that are in the collisional regime but transition to the collisionless regime when certain instabilities or plasma processes are triggered (e.g., solar flares).

To deal with this multi-scale challenge, most studies have either used fluid simulations, where kinetic scales are neglected, or used small kinetic simulations restricted to two-dimension. However, the advent of petascale supercomputers together with advances in computational codes are enabling the first glimpse of the physics of these processes in 3D as well as large 2D systems spanning a wide range of scales.

9.1.1.2 Scientific Objectives for 2017

The overarching scientific objectives are (a) to develop a more complete understanding of the three fundamental plasma physics processes (reconnection, turbulence, dynamo), and (b) use the gained knowledge to develop a more accurate forecasting of space weather. The Earth's magnetic field shields us for the most part from the solar wind, consisting of a magnetized plasma emanating from the Sun. Due to magnetic reconnection between the magnetic field in the solar wind and the magnetosphere, however, a small fraction of the solar wind penetrates the Earth's protective shield. During severe solar storms, billions of tons of superhot plasma are unleashed and can wreak havoc on our technological systems. It is estimated that a solar storm of the magnitude of the 1859 solar superstorm would cause over $2 trillion in damage today. In the face of this continual threat, there is an urgent need to develop accurate “space weather” forecasting capabilities. However, there are no reliable forecasting models that can predict either the severity or the location of the impact on Earth for a given solar storm. One key factor contributing to this poor performance is

the lack of appropriate models of reconnection in the global space weather codes. The development of such models is one of the primary objectives of our research.

9.1.2 Computational Strategies (now and in 2017)

9.1.2.1 Approach

The problems that we are working on require exascale-class computing and beyond. Thus we are continually pushing the state-of-the-art in simulations to take full advantage of the new computational capabilities as they become available. In order to deal with the extreme multi-scale nature of these physical processes, we use a two-pronged approach. We use 3D global hybrid (fluid electrons, kinetic ions) simulations to model the interaction of the solar wind with the Earth's magnetosphere. And we use large-scale fully kinetic simulations to study the effects of electron kinetic effects on large scale dynamics. Our largest simulation to date includes 10 trillion particles, more than 3 orders of magnitude larger than our largest simulations in 2006.

9.1.2.2 Codes and Algorithms

VPIC: a highly optimized 3D fully kinetic particle-in-cell (PIC) code, which solves the relativistic Vlasov-Maxwell equations

H3D: a petascale 3D global hybrid simulation code which treats the electrons as fluid but retains the full ion kinetic effects

PSC: a development fully kinetic code for testing out various algorithms and architectures. This code has been adapted to work on GPUs.

HYPERS: the first ever particle code with asynchronous updating of particles and fields based on discrete event technology.

9.1.3 HPC Resources Used Today

9.1.3.1 Computational Hours

We have allocations on Kraken (7 million hours), Jaguar/Titan (20 million hours), Blue Waters (30 million hours), and Pleiades (10 million hours). We used 4.5 million hours at NERSC in 2012.

9.1.3.2 Data and I/O

We try to manage with limited storage resources for most of our work, e.g., 2-3 TB of scratch (temporary) space, 2-3 TB of NGF project space and ~10TB of HPSS. But some of our cutting-edge results at NERSC (involving a 2 Trillion particle simulation in collaboration with the ExaHDF5 team) requires ~300-400 TB of scratch space. We would like to archive a reasonable fraction (~10s of TBs of data) from such runs.

We have been using Hopper, so we have not been sharing among NERSC systems.

The scratch and permanent space available to us are too small since even a single run can generate over 300 TB of data. Since the total disk space for our account is fixed and we have several people working under the same account, we have to ask for disk allocation for each

individual researcher to be switched back and forth to the maximum allowed for the account, depending who happens to be analyzing the data.

Our code uses a combination of custom binary output formats and HDF5 files. In collaboration with the ExaHDF5 project, we are exploring the use of collective writes to single, shared HDF5 files for storing our data; followed by indexing and querying. Both out custom output function and HDF5 library make extensive use of MPI I/O. Analysis of trillion particle datasets poses a number of challenges: we have utilized the FastQuery software for indexing and querying the data in a reasonable amount of time.

We use MPI I/O (both via HDF5 and raw calls). Efficient I/O and MPI-IO implementations are critical for our simulations. Due to the large number of particles in our simulations, the checkpointing files are very large (~60 TB per snapshot). Currently, it takes us ~1 hour to write checkpointing files, which is too long. Disk space and I/O bandwidth constrains us to write a snapshot once every ~2,000 timesteps; in an ideal world we would dump data more frequently. We have had difficulty on Hopper at times and other machines using Lustre with checkpointing (e.g., if there is a lot of contention on the disk). Some of our weak scaling tests in collaboration with the ExaHDF5 team have revealed that our application can utilize all available I/O bandwidth on hopper with a properly tuned stack. As NERSC upgrades its compute horsepower in the future, we are assuming that I/O resources will scale up accordingly, or else it will become infeasible for us to perform checkpointing and execute data dumps in a reasonable amount of time.

9.1.3.3 Parallelism

As mentioned, the problems that we are working on require exascale and beyond. The number of particles that we can fit on the memory limits the size of simulations and the type of scientific questions that we can address. As a result, most of our runs use the maximum number of available cores on Blue Waters, Titan and Kraken. Our codes scale nearly perfect with the number of cores. Hopper, however, does not cater to full machine jobs. The limitations in disk storage, user queues, etc. seem more targeted towards smaller jobs (below 50,000 cores). As a result, we have used Hopper mostly for smaller "test" runs and analysis of data from petascale runs. We should note that the analysis of massive data sets resulting from our runs require significant number of cores (1,000-2,000 cores) and can be very CPU intensive.

9.1.4 HPC Requirements in 2017

9.1.4.1 Computational Hours Needed

Since we are limited by the maximum available memory on a given system, our job profiles follow a simple formula. If we define the maximum number of cores as N, our typical job would require 72 hours of compute time, and we need about 6-10 such runs a year for a given research topic. As such, we plan to apply for allocations from other sources such as BlueWaters, Kraken, Titan, and Pleiades. The primary factor driving the need for more hours is pushing to larger simulations as the maximum number of available cores increases in time.

9.1.4.2 Data and I/O

Our disk/memory requirements scale up linearly with the total number of cores. Here we provide our requirements today. The requirements for 2017 would be a factor of 32X

larger. There are 8 arrays associated with each particle, and each takes 4 bytes. For our largest run today, including 10 trillion particles on 300,000 cores, this translates to 0.32 PB of data, not including the grid quantities. We typically save the particle data twice for a given run and the grid data about 60 times. We are working on in situ visualization as a way to save certain diagnostics at much higher time resolution. A single run of this size requires nearly 1 PB of disk, and more than ~0.6 PB of memory. Ideally, we would like to checkpoint data from memory to disk in ~30-45 minutes. Our disk/memory requirements would scale up linearly with the total number of cores.

9.1.4.3 Scientific Achievements with 32X Current Resources

There are many problems that are out of our reach currently due to the limitations in the total memory available on the largest computers available today. A factor of 32X increase in computational power/memory would enable us to address important scientific questions for the first time. For example, our recent study, using 2D simulations at realistic mass ratio, has shown that the structure of the electron layers does show strong dependency on the mass ratio. Similarly, some instabilities such as lower hybrid drift instability require high mass ratios. We cannot currently conduct large-scale 3D fully kinetic simulations of reconnection or turbulence at such mass ratios.

9.1.4.4 Parallelism

Our codes scale nearly perfectly with the number of cores. Thus we can use as many cores as will be available in 2017. Since the number of GPUs is expected to be much less than the number of cores, we expect to use GPUs for in situ visualization. Our smaller runs can run concurrently.

9.1.4.5 Memory

As mentioned, the size of our runs is determined by how many particles we can fit in memory. Each particle takes 32 bytes. The aggregate memory sets how many particles we can fit (which is a measure of the size of the simulation), and the number of nodes sets the limit on the number of time steps that we can run the code. For high mass ratio runs, this becomes a critical factor. Currently the largest simulations we can make are about 10 trillion particles. A factor of 32X increase in memory and number of nodes would allow us to make simulations with over 320 trillion particles. Such capability will allow us to study many science questions that are currently out of our reach.

9.1.4.6 Many-Core and/or GPU Architectures

We have a development code (PSC) that we have used for experimenting with GPUs and other technologies (e.g., Intel's MIC architecture). We expect our codes to be able to make use of hardware accelerators by 2017.

9.1.4.7 Software Applications and Tools

Fault tolerance hardware/software solutions would be quite valuable. I/O middleware solutions that can automatically tune I/O parameters (e.g. optimal striping, MPI-IO aggregators, etc.) would be quite desirable. We expect to continue our collaboration with the ExaHDF5 team to explore HDF5 and FastQuery tools for storing and analyzing results from our leading-edge simulations. Parallel interactive, batch, and in-situ visualization tools such as ParaView are essential today and will be into the foreseeable future. For instance ParaView in addition to traditional parallel interactive capability provides in-situ data analysis and reduction capabilities that will be necessary in extremely large simulation

runs. For post-run analysis and visualization it will be important to have a large ratio of ram per core available. As visualization software adapts to many-core architectures this will be important during since visualization codes have different scaling characteristics than do simulations.

We use gridFTP and Globus online for data transfer to Hopper from other machines. Since we are interested in bit-wise reproducibility in our simulations, we also need Intel compiler which has a flag that helps in this regard.

9.1.4.8 HPC Services

Support in data analysis and visualization, Parallel I/O, data sharing technologies, and collaborative tools are critical.

9.1.4.9 Time to Solution and Throughput

Having a point of contact is critical for making full-scale runs.

9.1.4.10 Data Intensive Needs

Data sharing, archiving and retrieval, and visualization remain a challenge.

9.1.4.11 What Else?

Currently Hopper is not optimally set up for full-machine runs. This is true both in terms of running the simulations at scale as well as storage, handling and analysis of the resulting data. In our work with the ExaHDF5 team, we have had to request NERSC staff for scheduling full-scale 120,000 core runs; which were rather hard to schedule and troubleshoot. We were able to do these runs only once in a 2 or 4-week period, which extended the science project over a period of 6-9 months. The ExaHDF5 team had to ask for repeated extensions for a 500TB disk quota on NERSC scratch system. We believe that it is challenging for routine NERSC users to push full-scale I/O intensive jobs through on hopper, without hand holding from NERSC staff. We are, of course, very grateful for the support we have received from NERSC staff in enabling our runs, but NERSC might need to consider a different strategy for full-scale runs.

Draft

9.1.4.12 Requirements Summary Worksheet

We have filled out this table assuming that we can make full-scale runs in repo m148 by 2017.

	Used at NERSC in 2012	**Needed at NERSC in 2017**
Computational Hours (Hopper core-hour equivalent)	4.5 Million	4.147 Billion[1]
Scratch storage	3 TB	32 PB
Shared global storage (/project)	1TB	64 PB
Archival storage (HPSS)	73 TB	640 PB
Number of cores* used for production runs	< 50,000 except one run at 100,000	Maximum available
Aggregate memory	400-1000TB	32PB

[1] 32x300 000(cores)*72(hours)*6

Draft

10 High Energy Density Laboratory Plasmas (HEDLP) & Inertial Fusion Energy Sciences (IFES) Case Studies

10.1 Heavy Ion Beams and Interactions with Plasmas and Targets (HEDLP and IFES)

Principal Investigator: Alex Friedman, LLNL and LBNL
Case Authors: A. Friedman, R. H. Cohen, D. P. Grote, W. M. Sharp (LLNL and LBNL); I. D. Kaganovich (PPPL); A. E. Koniges and W. Liu (LBNL/NERSC)
NERSC Repositories:
mp42 ("Simulation of intense beams for heavy-ion-fusion science," Alex Friedman, PI)
m74 ("Nonlinear δf Particle Simulation of Collective Effects for Heavy Ion Fusion Drivers and High Intensity Particle Accelerators," Ronald Davidson, PPPL, PI)

10.1.1 Project Description

10.1.1.1 Overview and Context

DOE's Fusion Energy Sciences program supports research on the science of intense heavy-ion beams, on high-energy-density physics (HEDP) and especially Warm Dense Matter (WDM) generated by ion beams, and on target physics for ion-beam-driven Inertial Fusion Energy. The major participants in this endeavor are the partners in the Heavy Ion Fusion Science Virtual National Laboratory (HIFS-VNL), a collaboration of LBNL, LLNL, and PPPL, and colleagues at universities. Ongoing experiments are focused on generating, compressing, and focusing intense, space-charge-dominated ion beams and using them to heat thin foils, the evolution and properties of which are then measured. To further this work, a new accelerator facility, the Neutralized Drift Compression Experiment-II (NDCX-II), has been built at LBNL, and is currently undergoing commissioning. Obtaining maximum benefit from these experiments is a key near-term goal of the simulation program.

Simulation efforts in support of NDCX-II have concentrated on developing the physics and engineering design of the accelerator, on identifying favorable operating points, and on planning the Warm Dense Matter experiments to be done with its beam once full operations ensue. To support machine operations and user experiments, the scope of our simulations is evolving, with primary emphasis on detailed simulations of the actual beams as realized and of their interactions with various targets. This includes extensive studies of the coupled dynamics of the beam and a neutralizing plasma (which allows the beam to be compressed into a compact volume in space and time). These studies employ the beam dynamics code Warp and other kinetic models, run primarily at NERSC. As NDCX-II shifts emphasis toward studies of the target response, the simulated ion beam data at the target plane will be transferred into hydrodynamic simulations of targets using the Hydra code (run at LLNL) and the ALE-AMR code (run at NERSC).

Intense ion beams are non-neutral plasmas and exhibit collective, nonlinear dynamics that must be understood using the kinetic methods of plasma physics. This physics is rich and subtle: a wide range in spatial and temporal scales is involved, and effects associated with

instabilities and non-ideal processes must be understood. In addition, multispecies effects must be modeled during the interaction of the beams with any stray electrons in the accelerator and with the neutralizing plasma in the target chamber. The models must account for the complex set of interactions among the various species, with the walls, and with the applied and self-fields. Finally, oscillations of the beam core and "mismatches" of the beam confinement can dilute the beam phase space and parametrically pump particles into a low-density outlying "halo" population; this physics imposes stringent requirements on numerical noise, requiring good particle statistics and mesh resolution. A blend of numerical techniques, centered around the Particle-In-Cell method, are used in Warp to address these needs, including: electrostatic and electromagnetic solvers, adaptive mesh refinement, cut-cell boundaries, a large time step particle pusher, and implicit field solvers and particle pushers. Ion beams have a long memory, and initialization of a simulation at mid-system with an idealized particle distribution is often unsatisfactory; thus, a key goal is to further develop and extensively exploit an integrated and detailed source-to-target beam simulation capability.

In order to determine material properties (such as equation of state) in the warm dense matter (WDM) regime, we must use simulation codes to help interpret experimental diagnostics. The WDM regime is at the boundaries of solid-state physics and plasma physics, between non-degenerate and degenerate material, between ionized and neutral states, and between liquid and vapor. To understand and simulate the results of experiments that volumetrically heat material, many processes must be combined into a single simulation. These include phase changes, ionization processes, shock processes, spall processes, and droplet formation, to name a few. One goal is to simulate as effectively as possible the experiments that will be conducted on NDCX-II, as it compresses ion beams and uses them to heat foil targets to temperatures of about 10,000 K. The matter is heated so rapidly that, although its temperature is well above that of vaporization, inertia will keep it at solid density (at least for an inertial confinement time of order one ns). Experimental diagnostics will record the response of the material (e.g. temperature, density, velocity) to infer the equation of state and other properties. Synthetic diagnostics in the hydrodynamic simulations will be essential for inter-comparison.

10.1.1.2 Scientific Objectives for 2017

Between now and 2017, the project will concentrate on using advanced simulations to provide critical support to the NDCX-II facility, including both its operations and the experiments that it will drive. There is considerable scope for optimization of the beam dynamics in the accelerator, in the neutralized drift line, and in the final transverse focusing system. Already, "ensemble" runs at NERSC have enabled beam optimization, but these have not included first-principles plasma models; by 2017 this should be routine.

NDCX-II was designed to be extensible and reconfigurable; on that timescale, the machine should be extended by the addition of ten additional acceleration cells (which are already available but require modification and installation), significantly increasing the beam kinetic energy and decreasing its pulse duration. This configuration, too, will need extensive simulation, including studies of the neutralization process and of non-ideal effects. Options for multi-pulse operation also need to be thoroughly explored, for "pump-probe" experiments.

Hydrodynamic codes such as Hydra and ALE-AMR can be used to model ion deposition and the subsequent response of the target to volumetric heating. In support of HEDLP experiments driven by NDCX-II, a core research effort will conduct hydrodynamic simulations with realistic incoming beams (obtained from both simulations and detailed

diagnostics, requiring reconciliation). NDCX-II is a user facility; thus, while some experiments will be carried out by the core group and its close collaborators, more will be fielded by visiting users. Almost all experiments will need at least some local simulation support, even when outside users employ their own codes as their principal simulation tools. LBNL, LLNL and SLAC have recently formed a Bay Area High Energy Density Sciences (BA-HEDS) cooperative to facilitate coordinated experiments, some driven by NDCX-II's ion beam, and others by high-power lasers and X-ray FEL's at the other labs. These developments will enlarge the scope of the simulation program.

The project will also carry out research on elements of target physics for heavy-ion beam-driven IFE, including beam-target energy coupling, new ideas for improved targets, and assessments of target behavior. While much of the actual target design work will be carried out on computer facilities other than NERSC (especially systems at LLNL), it will be important to employ NERSC for other aspects of this program element.

10.1.2 Computational Strategies

10.1.2.1 Approach

For each of the project goals described above, the strategies to be employed are:

(1) Optimize the properties of the NDCX-II beam for each class of target experiments; achieve quantitative agreement with measurements; develop improved machine configurations and operating points. To accomplish these goals, we plan to use Warp to simulate NDCX-II from source to target, in full kinetic detail, including first-principles modeling of beam neutralization by plasma. Additional tools, including kinetic codes such as LSP and BEST, will be employed as appropriate. The output from an ensemble of Warp runs (representing shot-to-shot variations) will be used as input to target simulations using ALE-AMR on NERSC, and other codes on NERSC and elsewhere.

(2) Develop enhanced configurations of NDCX-II, and carry out studies to enable a next-step ion beam facility (IB-HEDPF). To accomplish these goals, much of the work will involve iterative optimizations employing Warp. These will, at first, assume ideal beam neutralization downstream of the accelerator, but will then advance to first-principles plasma models.

(3) Carry out detailed target simulations in the Warm Dense Matter regime using the ALE-AMR code, including surface tension effects, liquid-vapor coexistence, and accurate models of both the driving beam and the target geometry. For this we will need to make multiple runs (to capture shot-to-shot variations), and to both develop and employ synthetic diagnostics (to enable comparison with experiments). The new science that will be revealed is the physics of the transition from the liquid to vapor state of a volumetrically superheated material, wherein droplets are formed, and wherein phase transitions, surface tension and hydrodynamics all play significant roles in the dynamics. These simulations will enable calculations of equation of state and other material properties, and will also be of interest for their illumination of the science of droplet formation.

10.1.2.2 Codes and Algorithms

Our main ion-beam code, Warp, was originally developed to simulate space-charge-dominated beam dynamics in induction accelerators for heavy-ion fusion (HIF). In recent years, the physics models in the code have been generalized, so that Warp can model beam injection, complicated boundary conditions, denser plasmas, a wide variety of accelerator "lattice" components, and the non-ideal physics of beams interacting with walls and

plasmas. The code now has an international user base and is being applied to projects both within and far removed from the HIF community.

Warp uses a flexible multi-species particle-in-cell model to describe beam dynamics and the electrostatic or electromagnetic fields in particle accelerators. While the core routines of Warp solve finite-difference representations of the field equations and relativistic or non-relativistic motion equations, the code also uses a large collection of subordinate models to describe lattice elements and such physical processes as beam injection, desorption, and ionization. The representation of particles by a much smaller number of "macroparticles" can be derived from Boltzmann's equation, describing the evolution of a population of particles interacting by collisions and the collective fields.

Warp is a 3-D time-dependent multiple-species particle-in-cell (PIC) code, with the addition of a “warped-coordinate” particle advance to treat particles in a curved beam pipe. Self-fields are obtained via Poisson equations for scalar and vector potentials, or via Maxwell equations. Time-dependent applied external fields can be specified through the Python user interface. Warp also has 2-D models, using Cartesian or cylindrical geometry, as well as low-order moment equations. Models are available for background gas, wall effects, stray electrons, space-charge-limited and source-limited emission, and atomic processes such as charge exchange. Elaborate initialization and run-time options allow realistic modeling of complex systems. A beam may be initialized with one of many analytic distributions or with a distribution synthesized from experimental data, or ions can be emitted from a flat or curved diode surface. Lattice-element fields may be represented by several options, from simple hard-edge analytic forms to first-principles 3-D calculations. Poisson's equation can be solved using several methods, including FFT, Multigrid, and AMR/Multigrid. The electromagnetic (EM) solver can also use MR. With multigrid, the Shortley-Weller method for the subgrid-resolution description of conductors allows the use of complicated boundary conditions.

Parallelization of Warp is done using domain decomposition with MPI. Warp uses independent spatial decompositions for particles and field quantities, allowing the particle and field advances to be load-balanced independently. In transverse-slice 2-D runs, the field solution is repeated on each node, but solved in parallel by processors within a node.

The size and duration of Warp jobs varies tremendously, depending on such factors as problem dimensionality, grid size, duration, particle count, and the physical processes being modeled. With our generalized decomposition, we do not foresee any limitation resulting from the code's architecture; but Poisson solution scaling is poor at large problem sizes. For a 3-D test problem using 512 x 512 x 512 cells, we have demonstrated excellent parallel scaling of the electromagnetic PIC capability, up to about 50,000 processors.

Our Warp projects tend not to be data intensive; they use modest amounts of memory but require many time steps. We typically run (in 2-D or 3-D) with of order 100 grid cells along each transverse axis and 1000 grid cells along the longitudinal axis. Large 3-D simulations typically have a mesh of order several 100s by 100s by 1,000s of grid cells. The data per cell is either several scalars or several 3-D vectors, depending on the field model. Typically of order 10^6 particles are used in runs without plasma electrons, with 13 or more variables per particle, and including dynamics electrons can require up to 30 x 10^6 particles. We currently use 120 to 1920 processors for typical Hopper runs and up to 6,144 for a few key runs with fine grids and an augmented number of particles.

ALE-AMR is a relatively new code that combines Arbitrary Lagrangian Eulerian (ALE) hydrodynamics with Adaptive Mesh Refinement (AMR) to connect the continuum to micro-structural regimes. The code is unique in its ability to model both hot radiating plasmas and

cold fragmenting solids. The hydrodynamics are done in a Lagrangian model (wherein material moves with the mesh), but the resulting mesh can be modified to prevent tangling or severe mesh distortions. If the entire mesh is restored to the mesh of the previous time-step after every step, the code is said to be run in Eulerian mode (fixed mesh). In general, this is not done, and we only modify a portion of the mesh during a fraction of the time steps. This ability to do selective remapping is the reason to use the word "arbitrary." We also employ the Hydra code, another 3-D radiation hydrodynamics ALE code; that code is run on LLNL computers, which are accessed from LBNL and LLNL by group members. A common feature of ALE codes is the ability to have multiple materials in a given computational zone. Such mixed zones are generally created during the advection phase of the advance, when material from the old mesh is transferred to the new mesh. The ALE-AMR code uses a volume-of-fluids approach to calculate the interface between different materials in a zone. Information from neighboring zones can be used to explicitly construct the interfaces if needed.

One key added capability of ALE-AMR, relative to other ALE codes such as Hydra, is the ability to dynamically add mesh elements (refinement) or remove mesh elements (coarsening) during the run. This capability is called Adaptive Mesh Refinement (AMR); however, the ability to remove zones by coarsening the mesh when there are no longer any steep gradients is also important. ALE-AMR refines by a factor of three along each dimension, so in 3D one zone becomes 27 zones. During refinement all material interfaces must be explicitly defined to place the correct amount of each material in the new zones. Numerical techniques were developed for many of the physics packages to work efficiency on a dynamically moving and adapting mesh. ALE-AMR also continues several features that allow for very long-time simulations, a unique fragmentation capability, and the ability to "shape-in" unusual objects.

Additional physics, beyond basic hydrodynamics, is implemented in ALE-AMR using operator splitting. For example, a flexible strength/failure framework allows "pluggable" material models to update the anisotropic stress tensor that is used in the hydro advance during the following step. The code also includes an ion-deposition model for bulk heating of the material, and both heat conduction and radiation transport using the diffusion approximation. The hydro uses explicit time stepping but some packages, e.g., radiation transport, can do an implicit solve at each explicit time step.

The parallelism in ALE-AMR is currently MPI-only with the ability to do dynamic load balancing based on the computational requirements. The domain decomposition is zonal. During the ion deposition phase of the simulation, the regions with ion beams will have smaller number of zones in the domain assigned to a given processor because of the additional computation work associated with beam energy deposition. There are various places in the code were additional levels of parallelism are possible and we are investigating hybrid models, e.g., OpenMP + MPI.

10.1.3 HPC Resources Used Today

10.1.3.1 Computational Hours

NERSC is (far and away) our principal computational resource. Computer resources at LLNL are used for HYDRA. To date these have been modest, but it is anticipated that this usage will expand as NDCX-II goes into production mode. Computational clusters at LBNL and PPPL also are employed for code development and testing, and for some of the same purposes as NERSC when the computational demands are much smaller.

10.1.3.2 Parallelism

The number of compute cores is highly problem dependent – Warp in particular is used for many different kids of runs. While we have used tens of thousands of cores, in a typical large run we use a few hundred to a few thousand. The largest number we would consider using in a production run, today, is 100,000. Fewer are typically used because the problems being solved do not require more. Also we need to conserve our allocation and minimize human effort, so for example, "ensemble" runs (many independent simulations in a single batch submission) use relatively small numbers of particles, at the price of jitter in the results. We sometimes have multiple jobs (ensemble or otherwise) running concurrently (up to about five). With regard to strong vs. weak scaling, this depends upon the field solver being used, e.g., the Poisson solver does not scale well to large numbers of cores, while the electromagnetic solver does. We want strong scaling for increased resolution, rather than for an enlarged problem domain. We are currently investigating the use of communication-reduced solvers in the context of Warp.

10.1.3.3 Data and I/O

Scratch (temporary) space: few 100 GB

Permanent (can be shared, NERSC Global Filesystem /project): 100 GB (including Home)

HPSS permanent archival storage: 1 TB

As noted above, our project does not tend to be storage intensive. During computations, data is primarily written to scratch, which we also use for short-term storage (few weeks) as runs are analyzed. For speed in the loading of Python (a topic of current interest), we keep Python installations in scratch (with different ones for dynamic and static loading, for example). We've been keeping our code repositories (for Warp and supporting tools) in /project. In Home, we keep smaller output data files for long-term reference and post-processing. HPSS is used for storing full output datasets, restart dumps, and large graphics files and data dumps.

I/O is of two general types: gathering data into the first processor and then writing it to disk; and direct writes by each processor into its own file.

With regard to constraints due to I/O, we have found that loading Python onto a large number of processors scales poorly. NERSC staff has helped us improve performance here, but it remains an area of concern for future machines in the 2017 time frame.

A typical run writes about 50GBs over a 24-hour period, so about 0.5MB/sec.

10.1.4 HPC Requirements in 2017

10.1.4.1 Computational Hours Needed

We anticipate a need for (of order) 40 M CRU's / year in the 2017 time frame, primarily driven by the need for computationally larger runs involving detailed plasma descriptions, by the need to run ensembles for optimization and sensitivity analysis, and by the desire to be able to use improved methods for optimization (large particle counts are needed in order to have a smooth optimization cost function). We also anticipate much greater use of the ALE-AMR code to support a wide variety of experiments on NDCX-II.

Our need for NERSC time is rapidly growing. In past years our requirements were modest. We concentrated on NDCX-II machine design using simplified tools such as a 1-D beam physics code (ASP), and on Warm Dense Matter simulations using both Hydra (at LLNL) and

a specialized 1-D code (Dish). In FY11, we began applying NERSC resources to iterative design calculations for the NDCX-II facility, and our usage rate increased roughly five-fold. Now, three developments compel us to carry out far more demanding simulations at NERSC: (1) the need to capture beam-in-plasma effects (requiring far more simulation particles, and a smaller time-step size and grid spacing); (2) Our interest in self-focusing of ion beams in plasma will require use of our explicit electromagnetic solver, which requires a substantially smaller time step due to the Courant condition; and (3) the introduction of target-response studies (requiring considerable resources for realistic problems).

We begin with a discussion of recent Warp usage, which has emphasized iterative design and assessment (on NERSC and LBNL clusters) using ensembles of runs with random errors. For this task 256 cases (instances) are typically run in a single batch job. The number of cores has ranged between 768 and 6144, depending on the problem, with less than 1 GB/core, using 60 GB total memory and 4 hours of wall-clock time. Much data processing is in-line, and I/O is only about 100 GB / batch job. This approach leads to very light traffic in and out of NERSC, with results stored at the Center.

Another class of Warp runs models ion beams in plasmas. Current problems of this type are axisymmetric runs using 100's x 1000's of cells and tens of millions of particles (with 13 or more variables per particle). These runs typically require 15-30 wall-clock hours on 120 processors of Hopper. With the Maxwell (EM) field model, tests show good scaling at fixed problem size (512^3 cells) to 50,000 processors (see Figure below).

We project the need for four classes of Warp runs: (1) Ensemble runs to optimize the output beam from the NDCX-II accelerator, for each class of target being shot. So far, we haven't used gradient methods for optimization because of particle noise; we hope to overcome this with larger runs. (2) Simulations of plasma injection into the drift-compression line and final-focus solenoid, which can be quite costly because the plasma flow is relatively slow (~10 ms) and it is necessary to operate on an electron timescale. Both EM and explicit electrostatic (ES) models are used; run times are comparable because the time-step size in the ES model, which is set by the need to resolve plasma oscillations, is often near the Courant limit for light waves on the mesh. Also, the EM algorithm scales more readily to very large numbers of processors. (3) Integrated simulations of one or more beams compressing in a neutralizing plasma (with properties obtained from plasma-injection runs as described above, or via measurements). Such runs require less computer time than plasma injection runs, because the beam is in the system for < 1 ms; however, ensembles are typically needed. (4) Detailed simulations resolving short time- and space-scales for, e.g., two-stream instability. Since the highest growth rates for a cold beam and plasma are for short wavelengths, while we seek to capture the overall system scale, such runs can be costly, even in axisymmetric (r,z) geometry. Projected 3-D runs will, of course, require substantially greater resources.

10.1.4.2 Parallelism

A typical run is expected to use tens of thousands of core, with large runs using hundreds of thousands. Ensemble runs (with many cases run concurrently in a singe submission) can, in principal, make good use of any number that are available to us.

We will definitely need multiple jobs running concurrently; up to 10,000 in an ensemble would be useful, but not strictly necessary. In non-ensemble cases, modest numbers of large runs are needed; that is, a similar total core count, but fewer simultaneous jobs.

10.1.4.3 Data and I/O

Scratch (temporary) space: few TB

Permanent (can be shared, NERSC Global Filesystem /project): 1 TB (including Home)

HPSS permanent archival storage: 40 TB

I/O Rates: we have only been bandwidth limited in the sense that loading the code onto many processors has been an issue, requiring special approaches (e.g., a static build of Python) that we would refer to avoid if possible.

All of these increased requirements derive from the greater problem sizes anticipated, and from a somewhat greater number if runs (factor of several)

10.1.4.4 Scientific Achievements with 32X Current Resources

Our projected increase in needs, as described above, is by somewhat more than a factor of 32. Thus the achievements described above are relevant here.

10.1.4.5 Memory

Per-core, we anticipate needs comparable to those of current-day runs, that is, of order 1 GB per core (number per node will depend upon the number of cores per node). Aggregate memory will increase commensurately with the number of cores.

10.1.4.6 Many-Core and/or GPU Architectures

We do not currently employ GPU's, but have been exploring the possibility. While the central loop of Warp could be readily adapted, end-case challenges (diagnostics, boundary conditions, etc.) make a full port to a GPU architecture challenging. We will rely on NERSC assistance to help us assess what is possible here, and to implement the best strategy.

10.1.4.7 Software Applications and Tools

We anticipate need for MPI, and the capability for MPI at the Python level. We need compilers for Fortran, C, C++, and perhaps OpenACC, OpenCL or something similar for accelerator-enabled code. Much of our I/O in Warp (e.g., dump files) is through Python, posing challenges for performance; some code restructuring may be needed to take advantage of MPI I/O.

10.1.4.8 HPC Services

We anticipate a need for consulting as the available NERSC resources evolve. Much of our visualization is in-line; CGM files are likely to remain important to our workflow. We hope to be able to make increased use of VisIt with Warp. ALE-AMR currently uses VisIt as its primary visualization tool. Our needs with regard to other services are not unusual.

10.1.4.9 Time to Solution and Throughput

We do not anticipate any unusual needs beyond the existing pattern of queue structures, limits, etc.

10.1.4.10 Data Intensive Needs

Larger Warp runs may ultimately drive some use of data-intensive computing techniques. We do not anticipate any special needs associated with data-intensive computing.

10.1.4.11 What Else?

Interactivity (on the main NERSC computer, for parallel computations) is essential for debugging and diagnostics development, because Warp produces many of its diagnostics on-line (thereby avoiding massive data transfers, offloading mostly processed data).

The Warp code is deeply tied to Python. Dynamic libraries are part of the core of Python and continued support for them would be useful. Dynamic libraries are not actually essential, since Warp and Python can be built statically, however, static loading requires modification of the Python source, introduces a more complicated and fragile build process, and makes installing new and upgrading existing packages more difficult since they must be manually incorporated into the build system (when they could otherwise be installed independently). Solving the outstanding issue of scalable and efficient loading of dynamic libraries would remove the need for the static build and would greatly ease code maintenance and preserve flexibility.

We will continue to have a need for scatter-add deposition of source terms from the particles onto the mesh. Integrity of these terms must be preserved as we optimize performance using (probably) MPI plus on-node parallelism, and AMR.

Draft

10.1.4.12 Requirements Summary

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours (Hopper core-hour equivalent)	0.4	40
Scratch storage	0.3 TB	3 TB
Maximum number of cores* that can be used for production runs	(not critical)	(not critical)
Shared global storage (/project)	0.1 TB	1TB
Checkpoint bandwidth	(not critical)	(not critical)
Archival storage (HPSS)	0.75 TB	40 TB
Maximum I/O bandwidth (excluding checkpoint data)	(not critical)	(not critical)
Number of conventional cores used for production runs	3000	100,000
Memory per node	1 GB	1 GB
Aggregate memory	3 TB	100 TB

Draft

10.2 Laser Plasma Interactions

Worksheet Authors: Chuang Ren (University of Rochester); W. B. Mori, F. S. Tsung (UCLA)

NERSC Repositories:

"Investigations of advanced ignition physics and extreme states of matter," m412, PI: W. B. Mori (6,641,650 used in 2012)

"Study of laser-plasma interactions relevant to direct-drive ICF implosions," m792, PI: C. Ren (7,767,525)

"2D and 3D simulations of SRS under IFE Relevant Conditions," m1157, PI: T.S. Tsung (5,422,459)

19,831,634 total hours

10.2.1 Project Description

10.2.1.1 Overview and Context

The need to develop a cheap, safe, and plentiful source of energy is obvious. One possibility is controlled fusion and one candidate is inertial fusion energy (IFE). In laser driven IFE, laser energy has to be effectively coupled to a target to drive a symmetric implosion with high-speed. During the implosion, the fuel shell has to maintain a low-entropy state and hydrodynamic instabilities have to be controlled to prevent shell breakup. Laser plasma instabilities (LPI) are central to many of these interconnected challenges. In both indirect- and direct-drive schemes, LPI are unavoidable. They can reduce the amount of energy coupled to a target through backscattering. They can affect implosion symmetry through crossbeam energy transfer. They can raise the shell entropy through preheating from suprathermal electrons. Understanding and predicting LPI are vital for the success of ICF. More broadly, lasers are also a primary driver to create high-energy-density (HED) plasmas in laboratory. Understanding how laser energy couples to targets is a critical part in any laser-driven HED experiment.

The overall goal of our research is to develop a first principle-based understanding of LPI and other HED physics that can be included in design codes in a computation-efficient way. Particle-in-Cell (PIC) simulations play a unique role in developing such an understanding for this highly nonlinear and kinetic physics. Focused and in-depth PIC simulations can identify the essential physics of a particular physical process. They can also provide benchmark to ensure the physics to be properly included in a particular aspect of design codes. Due to the intense computation required, PIC simulations are at the frontier of high-performance computing, requiring state-of-the-art computation facilities and development of new algorithms for new computer architectures.

In this case study, we will use stimulated Raman scattering (SRS), two-plasmon-decay (TPD), and hot electron-assisted shock generation as examples to illustrate the computation needs in IFE/HED in 2017. They are chosen because they are fundamental issues in both indirect and direct-drive IFE, including an advance ignition scheme: shock ignition. They are chosen also because they require unprecedented computation resources to make progress in these intrinsically three-dimensional and/or multi-scale physics.

10.2.1.2 Scientific Objectives for 2017

1. Understand SRS and TPD in a laser beam of many speckles

An actual laser beam in ICF consists of many speckles, formed by breaking a coherent wavefront using distributive phase plates. With polarization smoothing, laser polarization can also change from speckle to speckle and even within a single speckle. SRS and TPD evolution and hot electron generation in such a beam are very different from those in a plane wave and intrinsically 3D. The speckles are coupled via the plasma waves and hot electrons generated. The state-of-the-art PIC simulations comprised 2 speckles in 3D, much less than the tens of thousands of speckles in an actual beam. A goal in 2017 is to study SRS and TPD with 3D PIC simulations comprising of ~100 speckles. These simulations are expected to close the gap between current 2D plane wave simulations and the experiments on observables such as hot electron generation.

2. Understand hot electron-assisted shock generation in shock ignition

Shock ignition is a new ignition scheme where a target is first compressed by an assembly pulse and then ignited by a high intensity spike. The assembly pulse is to compress the fuel to high density without creating a hot spot. Ignition is achieved by a high intensity spike launched at the end of the assembly pulse to drive a Gigabar ignitor shock that would collide with the return shock and create ignition conditions near the inner surface of the assembled fuel. Without needing to create the hot spot, the implosion in shock ignition can be on a lower velocity curve, thus reducing the driver energy and hydrodynamic instabilities and leading to a more robust and higher-gain ignition. At the high spike intensity, SRS and TPD are expected to generate significant amount of hot electrons. Whether these electrons can be stopped at the surface of the compressed shell to generate a strong shock is a critical issue is shock ignition research. To study electrons transport in high-density plasmas on the hydrodynamic time scale poses a grand challenge for PIC simulations. Hybrid PIC codes where a reduced set of Maxwell's equations using an Ohm's law is solved in the high density region can be an efficient tool for this problem, provided they are properly benchmarked against full PIC simulations. These simulations will help either validate or disqualify the shock ignition concept.

10.2.2 Computational Strategies (now and in 2017)

10.2.2.1 Approach

Our main challenge is the intense computation required and the limited resource available. A 3D simulation of 100 interacting laser speckles requires ~180 million processor hours, which is 12.5 times the total hours we used in 2012. Our codes have already achieved excellent scalability, up to 1.5 million processors (on the LLNL Sequoia system). Constrained by the available resources, we scaled down system sizes, dimensions, and number of particles used, which can all affect simulation fidelity. We continue to develop new algorithms to take advantage of new platforms such as GPU and also new reduced-description models.

10.2.2.2 Codes and Algorithms

Particle-in-Cell (PIC) codes model plasmas as particles that interact self-consistently via the electromagnetic fields they themselves produce. These models work at the most fundamental, microscopic level. As a result, they are the most compute intensive model in plasma physics. PIC codes generally have three important procedures in the main iteration loop. The first is the deposit, where some particle quantity, such as a charge, is accumulated on a grid via interpolation to produce a source density. Various other quantities can also be deposited, such as current densities, depending on the model. The second important procedure is the field solver, which solves Maxwell's equation or a subset to obtain the electric and/or magnetic fields from the source densities. Finally, once the fields are obtained, the particle forces are found by interpolation from the grid, and the particle co-ordinates are updated, using Newton's second law and the Lorentz force. The particle processing parts dominate over the field solving parts in a typical PIC application. OSIRIS uses finite difference for field solve and particle pusher. Parallelization is done using domain decomposition with MPI across nodes and it has an option for using shared memory parallelization using OpenMP within a node.

10.2.3 HPC Resources Used Today

10.2.3.1 Computational Hours

We used about 19.8 million processor hours in 2012.

10.2.3.2 Data and I/O

We mainly use scratch area (~12 TB) to store restart files and HPSS for archival data. We did not use /project space in 2012 although in the past we used it. For large simulations, OSIRIS has parallel I/O based on hdf5. It also has the option of run-time or post-process file merging. So far, in production runs with up to 20k cores, we have not had problems with I/O. We estimate having about 70 TB of HPSS storage.

10.2.3.3 Parallelism

We typically used 2,000 – 20,000 cores. Our codes currently scale up to 1.5 million cores. The number of cores we actually used was limited mainly by availability, i.e. the queue turnaround time. We would like to have more jobs running concurrently but each user typically have one job running per the current NERSC policy. For us weak scaling is more important than strong scaling.

10.2.4 HPC Requirements in 2017

10.2.4.1 Computational Hours Needed

We anticipate requiring ~550 M hours. The need is driven by having to include a fair amount of speckles in 3D to simulate laser absorption and hot electron generation in realistic beam settings. This is essential in establishing a predictive capability in LPI since the 2D plane wave simulations so far overestimated, for example, TPD hot electron generation by a factor of 5-50. Each 3D simulation with 100 speckles requires 140 M hours. We anticipate to do 1-3 such high fidelity runs together with other smaller scale exploratory runs in 2017.

10.2.4.2 Data and I/O

We expect a 20-30x increase of storage needs, in proportion to the processor hours increase. We can accept the current I/O rates, although an increase in the I/O rates would devote more computer time to computation. An estimate of 50 GB/sec is needed for completing checkpoint file writing within 30 minutes.

10.2.4.3 Scientific Achievements with 32X Current Resources

In 2017, with 32x current resources, our PIC simulations are expected to simulate LPI in ICF under realistic laser and plasma conditions and can compare with experimental results. We have every reason to believe our codes will be ready to take advantage of the increase resources in 2017.

10.2.4.4 Parallelism

Our runs in 2012 routinely used 20k cores and this usage was limited only by the cores available to us. We can easily use 10x more cores. We expect to run one, at most two, such jobs at a time.

10.2.4.5 Memory

Ideally, we typically would like 1 GB/core. This number is expected to stay constant in 2017 since our job size increases typically through weak scaling.

10.2.4.6 Many-Core and/or GPU Architectures

We have already developed a parallel GPU version of OSIRIS that has been tested on the UCLA Dawson cluster and shown to scale well to ~300 GPU's. Speed-ups of ~40 from the SSE version of OSIRIS and ~80 from the non-SSE version on the fastest i7 processor have already been achieved for linear current deposition. Currently we continue to develop GPU algorithms for high-order current deposition and collisions. Larger speed-ups are expected for higher order particle interpolation. We believe we will be ready to take advantage of a large GPU-assisted cluster in 2017.

10.2.4.7 Requirements Summary

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours (Hopper core-hour equivalent)	19.8 M	550 M
Scratch storage and bandwidth	12 TB	40-300 TB 50 GB/sec
Shared global storage (/project)	3 TB	20 TB
Archival storage (HPSS)	79 TB	1,400 TB
Number of cores* used for production runs	20,000	~200,000
Memory per node	1 GB	1 GB
Aggregate memory	20 TB	200 TB

11 Materials and Plasma-Surface Interactions

11.1 Plasma Materials Interactions

Principal Investigator: Brian D. Wirth, University of Tennessee
NERSC Repositories:
m1200: Computational modeling of plasma - surface interactions in tungsten exposed to mixed He-H plasmas
m916: Ab-initio modeling of the energetics and structure of nanoscale Y-Ti-O cluster precipitates in ferritic alloys
m1709: PSI SciDAC: Bridging from the Surface to the Micron Frontier

11.1.1 Project Description

11.1.1.1 Overview and Context

Plasma-material interactions pose an immense scientific challenge and are one of the most critical issues in magnetic confinement fusion research. The demands on plasma-facing materials in a steady-state fusion device include extreme particle and thermal fluxes. These energetic fluxes have pronounced impacts on the topology and chemistry of the near-surface region of the material, which influence the plasma sheath potentials and subsequent incident particle flux spectra. These evolutions are also inherently multiscale in time and are likely controlled by diffusional phenomena that are influenced by the high heat loads and subsequent thermal (and stress) gradients in the material, as well as by defect micro/nanostructures induced by both the ion and neutron particle irradiation. This complexity is further underscored by the fact that the plasma and materials surface are strongly coupled to each other, mediated by an electrostatic and magnetic sheath, despite the vastly different physical scales for surface (~ nm) versus plasma (~ mm) processes.

For example, the high probability (> 90%) of prompt local ionization and re-deposition for sputtered material atoms means that surface material in contact with the plasma is itself a plasma-deposited surface, not the original ordered material and an unintended alloy at that. Likewise, the recycling of hydrogenic plasma fuel is self-regulated through recycling processes involving the near-surface fuel transport in the material and the ionization sink action of the plasma. Another very serious aspect of the plasma-wall radiation damage in fusion reactors is tritium retention. If any carbon is present in the reactor, it will erode in the form of small CH radicals and molecules, which in turn tend to stick in other parts of the reactor, forming both soft and hard carbon films. These can have very large T contents in a form, which is hard to remove, and this has in fact been the major reason why the revised ITER design projects that carbon-based materials will not be used in the first wall during D+T operation. Furthermore, the intense radiation environment (ions, neutrons, photons) experienced by materials exposed to the fusion plasma environment ensures that the material properties are modified and dynamically coupled to the plasma materials surface interaction processes. Some of the most critical plasma materials interaction issues include: i) the net erosion of plasma-facing surfaces; ii) net tritium fuel retention in surfaces; iii) H isotope and material mixing in the wall; and iv) the minimization of core plasma impurities. Furthermore, the plasma-material surface boundary plays a central role in determining the fusion performance of the core plasma. However, while it is widely accepted that the plasma-surface interface sets a critical boundary condition for the fusion plasma, predictive capabilities for PSI remain highly inadequate.

Gaining understanding and predictive capabilities in this critical area will require addressing simultaneously complex and diverse physics occurring over a wide range of lengths (angstroms to meters) and times (femtoseconds to days and beyond to operating lifetimes). The lower time and length scales correspond to individual ion implantation and sputtering, which occurs at or near the material surface, in addition to a range of ionization and recombination processes of the sputtered neutrals and ions in the near surface sheath. At intermediate length and time scales, a wealth of physical processes are initiated, including diffusion of the now implanted ionic/neutral species, the possibility of chemical sputtering processes at the surface, the formation of gas bubbles, surface diffusion driving surface topology changes and phonon scattering by radiation defects that reduces the thermal conductivity of the material. At longer length and time scales, additional phenomena such as long-range material transport in the plasma, re-deposition of initially sputtered surface atoms, amorphous film growth and hydrogenic species diffusion into the bulk material and permeation become important. This broad palette of physical phenomena will require development not only of detailed physics models and computational strategies at each of these scales, but algorithms and methods to strongly couple them in a way that can be robustly validated. Furthermore, efficient and effective modeling techniques and frameworks need to be developed to couple the materials surface evolution and eroded material source terms into models of the sheath, scrape of layer and edge plasma physics. While present research is confined to each of these scales, or pioneering ways to couple two or more of them, the current approaches already push the state-of-the-art in technique and available computational power. Therefore, simulations spanning multiple scales needed for ITER, DEMO, etc., will require extreme-scale computing platforms and integrated physics and computer science advances.

11.1.1.2 Scientific Objectives for 2017

Successful development of PFC simulation tools will provide the ability to evaluate steady-state performance of tungsten-based PFC and divertor components in burning plasma environments. This will enable the identification of critical experiments to confirm whether practical PFC solutions exist for magnetic fusion energy beyond ITER.

The critical fusion materials science questions that will be able to be resolved through access to increasingly capable high-performance computing resources and PFC simulation tools are:

- What physical parameters control the time dependent evolution of the near-surface morphology and composition in the re-deposition layer (e.g., is this spatial re-deposition zone ever in a steady, or quasi-equilibrium, state?)—key phenomena required within predictive modeling include recycling, surface morphology, gas bubble, precipitate and second phase domains, and gas fueling/recycling;

- What are the effects of high-energy neutron damage on mediating, or exacerbating, near- surface defect evolution and tritium species permeation and retention;

- The impact of dilute impurities on surface morphology evolution and plasma contamination (e.g., Be in plasma, mixed material transport in tokamaks on erosion/impurity generation); and

- How does the evolving bulk microstructure impact the thermal properties, and thereby feedback into PFC evolution by modifying the resulting temperature profiles.

11.1.2 Computational Strategies (now and in 2017)

11.1.2.1 Approach

Addressing the critical fusion materials science questions associated with plasma surface interactions requires a number of research activities performed within a computational multiscale materials modeling paradigm. Our project is based on simultaneously attacking this problem from both a "bottom-up" atomistic-based approach, as well as from a "top-down" continuum perspective that focuses on kinetic models of species reactions and diffusion. The simultaneous approach improves the prospects for scale bridging, or multiscale integration, which is imperative in complex, inherently multiscale materials degradation such as plasma surface interactions.

Our project encompasses a full suite of multiscale materials modeling approaches, the current proposal is focused entirely on attacking the problem from an atomistic perspective using the LAMMPS molecular dynamics code, and a continuum, reaction – diffusion cluster dynamics modeling approach, using our PARASPACE code and the currently under development XOLOTL-PSI code.

The classical reaction-diffusion kinetic rate theory defines a defect/cluster by its character, atomic configuration and size (or, more specifically, the number of point defects contained within the cluster), but not by its spatial position since the theory assumes that the concentration of each cluster is homogeneous throughout the volume of the sample. However, more recent observations of radiation damage in irradiated materials, as well as spatially dependent ion implantation and plasma exposure of materials dictate the need for incorporating spatial dependence into the theory. PARASPACE is capable of handling multiple types of clusters, containing for example a combination of intrinsic defects (self-interstitial atoms or vacancies) and foreign gas atoms (helium, krypton etc.). PARASPACE serves as a working model for Xolotl-PSI development. Modeling PFC behavior with Xolotl-PSI requires extension of the number of cluster species to incorporate many more species (e.g., deuterium, tritium, helium, as well as vacancies, self-interstitials and impurity atoms such as Be or C that can induce separate chemical phases) within spatial domains.

The partial differential reaction-diffusion equation for each species *i* has the general form:

$$\frac{\partial C_i(\vec{r},t)}{\partial t} = \phi x P_i(\vec{r}) + \nabla \cdot [D_i \nabla C_i(\vec{r},t)] + GRT + GRE - ART - ARE \quad \textbf{(1.1)}$$

where Ci refers to the volumetric concentration of the i-th cluster at the spatial position, $\vec{r}$, ϕ is the incident gas particle flux, Pi is the production "probability" of the i-th cluster by irradiation (or the implantation probability for an injected gas or impurity atom) at the location $\vec{r}$, Di is the diffusivity of the i-th cluster, and ∇ denotes the gradient operator. The diffusion term can also be modified to account for biased-diffusion in a stress field. The rates of reaction amongst species that can lead to addition or subtraction of clusters of type i through a combination of 1st- and 2nd-order reaction kinetics, include GRT, the rate of generation of the i-th cluster by trapping reactions (A + B è i) among other clusters, GRE, the rate of generation of the i-th cluster by emission processes (C è i + B) of other clusters, ART, the rate of annihilation of the i-th cluster by its trapping reaction with other clusters (i + B è C), and ARE, the rate of annihilation of the i-th cluster by its own emission process (i èA + B).

The system of partial differential equations represented by the coupled equations defined by Eq. (1.1) are converted to ordinary differential equations by defining each cluster at

every spatial depth grid point and discretizing the spatial gradient terms. The resulting coupled ODEs include variation within the reaction terms to ensure point defect conservation, i.e., that all the reactants and products of any reaction (trapping or emission) are accounted for and are constrained within the prescribed phase space. The total number of ODEs is equal to the total number of clusters multiplied by the total number of depth grid points. It is not uncommon for simulations using the 1D PARASPACE code to reach O(107) degrees of freedom.

11.1.2.2 Codes and Algorithms

LAMMPS: Developed for large-scale molecular dynamics runs. Utilizes relatively short range (< 4 nm) interatomic potentials of N-body or pair interactions. Libraries requirements include MPI/MPICH. Web-site: http://lammps.sandia.gov

PARASPACE: Efficient one-dimensional reaction – diffusion cluster dynamics code. Utilizes both domain decomposition and threaded parallelization, and the PARDISO [29] code package for the implicit time integration of sparse, linear matrices. Libraries requirements include MPI/MPICH, OPEN-MP, PARADISO. Web-site: http://www.pardiso-project.org

XOLOTL-PSI: Spatially-dependent, finite element formulation of the reaction-diffusion cluster dynamics model under development as a built from scratch code supported by our SciDAC-PSI project. David Bernholdt and Jay Billings are currently developing and writing the code, and will be responsible for its use on NERSC HPC.

11.1.3 HPC Resources Used Today

11.1.3.1 Computational Hours

Since the basis for comparison in this review is Allocation Year 2012, we mention the two repositories that were active that year: m916 (1.7M hours used at NERSC) and m1200 (0.6M hours used at NERSC). Beginning in AY2013, the PI also has another NERSC project, m1709, usage for which will not be included here.

11.1.3.2 Data and I/O

Scratch (temporary) space: We typically use on the order of 30 GB in $SCRATCH to store checkpoint files within 6 weeks. Scratch space is used for restart (checkpoint) files, and dump files (the latter are backed up to permanent storage and/or local storage at intervals).

Permanent (can be shared, NERSC Global Filesystem /project): We typically use on the order of 5 GB in $HOME. The "permanent" space is output files, log files, code, executables, etc.

HPSS permanent archival storage: We typically transfer data files for visualization to our local computing cluster for visualization and archival storage, however as our MD simulations become larger in the future, we are likely to require additional archival storage.

We do not share data among machines currently, though Hopper and Edison will likely be data sharing in the future.

Checkpointing: We write checkpoint/restart files (~1.2 seconds each) every 4 minutes or so; recently reduced to every 12 minutes to increase efficiency (time lost repeating

calculations vs. time spent writing checkpoints).

11.1.3.3 Parallelism

Strong scaling is significantly more important to us than weak scaling. Maximum number of cores is 1200 cores (50 nodes) for "small" jobs (~4 million atoms) and 4080 cores (170 nodes) for "large" jobs (~16 million atoms). These numbers are, to my knowledge, near the "maximum" usable without significant parallel inefficiency, though we have plans to re-evaluate these numbers in the coming weeks. The frequency at which the jobs are scheduled is also a factor in the calculation of the number of cores.

At any time, user karlh typically has 7-8 "small" calculations (which may soon expand to 8-10) and one "large" calculation running. We also have three or four "large" calculations in the future, either at NERSC or at another facility, and would like to increase this number.

11.1.4 HPC Requirements in 2017

11.1.4.1 Computational Hours Needed

At current production rates, user karlh will use something like 20 million hours in 2014 on "small" simulations. If we move all calculations to NERSC facilities, that number could easily be more like 45 million hours (including a similar number of "large" calculations for comparison). By 2017 we anticipate needed 450 million hours for our runs.

Our need for more hours for large-scale MD of plasma-facing materials comes from the need to test different surface orientations, different helium plasma implantation fluxes, since we are currently perform simulations that are three to five orders of magnitude accelerated compared to experiments, and different spatial areas. Also, as the calculations run longer, we anticipate needing to add more atoms to the system, raising both the number of cores used and the amount of storage space (the amount of additional storage needed is relatively small).

11.1.4.2 Data and I/O

We generally experience bandwidth on the order of about 280 MB/s; in our worst-case example, this means approximately 2% of the time is spent on I/O, which is acceptable. Faster is obviously better, but not necessary.

11.1.4.3 Scientific Achievements with 32X Current Resources

Simulation of larger systems is anticipated, which should continue to decrease the simulated plasma implantation flux closer to experimental conditions, and this should be possible, as we anticipate near-ideal weak scaling. Longer times are our goal, however, so increases in strong scaling are preferable.

11.1.4.4 Parallelism

Using LAMMPS for molecular dynamics simulations we expect to be able to increase by about 20x in general core usage, although this could increase by 50-100X if we also can effectively utilize GPUs. A GPU version of LAMMPS is available and we are currently testing performance on TITAN at OLCF.

11.1.4.5 Memory

Memory requirements are modest; our largest calculation uses mem=10312kb,vmem=75528kb (used on each of 24 nodes, or 4080 cores), according to Hopper's reporting scheme. LAMMPS reports 21.5 GB, or 5.38 MB/core, as a lower-bound.

11.1.4.6 Many-Core and/or GPU Architectures

LAMMPS currently has GPU-based accelerator packages. We are in the process of testing those capabilities for our purposes at the OLCF's Titan computational facility, and will use them (both at NERSC and elsewhere) if there is an advantage to doing so.

11.1.4.7 Software Applications and Tools

LAMMPS is currently installed on NERSC machines, though the libraries for it (e.g., "make makelib; make -f Makefile.lib hopper") are not built---as such, I build them myself. It would also be nice to have a Python 3 interpreter for Hopper (one exists for Edison). Other than that, I have been quite satisfied with NERSC's available applications, especially the available compilers/interpreters for AWK, Python 2, Fortran (incl. 2003/2008), and C/C++.

11.1.4.8 HPC Services

We anticipate that as we utilize larger and larger MD simulations, we will be more interested in performing in-situ visualization and data compression at NERSC.

11.1.4.9 Requirements Summary Worksheet

	Used at NERSC in 2012	Needed at NERSC in 2017
Computational Hours	2.3 Million	450 Million
Scratch storage and bandwidth	0.05TB	10TB
	0.3GB/sec	250GB/sec
Shared global storage and bandwidth (/project)	0 TB	100TB
	N/A	250GB/sec
Archival storage and bandwidth (HPSS)	1TB	300TB
	0.5GB/sec	500GB/sec
Number of cores* used for production runs	4,080	64,000
Memory per node	100 GB	100 GB
Aggregate memory	2.4 TB	30 TB

Draft

12 Integrated Modeling

12.1 Integrated Whole-Device Modeling Studies

Principal Investigators: Alexei Y. Pankin, Tech-X Corporation, and Arnold H. Kritz, Lehigh University
Case Study Authors: Arnold H. Kritz, Alexei Y. Pankin, and Tariq Rafiq
NERSC Repositories: m1043 (A.Y. Pankin, PI); m649 (A.H. Kritz, PI)

12.1.1 Project Description

12.1.1.1 Overview and Content

The objective of this study is to investigate a possibility of integrated whole-device modeling (WDM) studies at NERSC and to evaluate the HPC requirements up to 2017. This study is complementary to the main research topics in the repositories m1043 (Modeling of tokamak plasmas with large scale instabilities) and m649 (Integrated modeling simulations for the plasma edge and core). The importance of the integrated WDM studies is related to specific physics problems for which the interactive nature of multi-scale physics is critical for understanding the nonlinear and nonlocal processes in tokamak plasmas. Typically, the interactive modeling codes include modules that describe the plasma heating, MHD equilibrium, large-scale instabilities, and core and edge transport. The WDM codes might also include modules for the scrape-off-layer region and plasma-wall interactions. The main challenges of the WDM studies are related to the fact that different components have different characteristic length and time scales and, consequently, different computation requirements. These two factors make optimal load balancing a very difficult task.

In this report, our experience with the TRANSP and FACETS codes is summarized. Other codes that were used in this study include the neoclassical kinetic XGC-0 code, the M3D-OMP equilibrium solver, and the ideal MHD stability ELITE code. These codes were coupled with the TRANSP and FACETS codes in order to describe specific physics effects or to model specific regions of tokamaks.

12.1.1.2 Scientific Objectives for 2017

Figure 1. Data flow and code coupling schema

The integrated WDM modeling codes are currently being used in the interpretive analysis of the experimental data and in the predictive studies of future tokamak discharges. The integrated WDM codes are also important for the verification and validation of physics models. A coupling schema in recent predictive modeling studies of transient flux effects on the H-mode pedestal stability is shown in Fig. 1. High physics fidelity codes and tools that have been developed during three SciDAC projects are used in this study. The FACETS code is an integrated whole-device predictive transport code that has been developed as a part of the Framework Application for Core-Edge Transport Simulations (FACETS) project (http://facetsproject.org/). The XGC0 code is a kinetic neoclassical code that has been developed as a part of the Center for Edge Physics Simulations (EPSI) project (http://www.cims.nyu.edu/cpes/). The Integrated Plasma Simulator (IPS) is a framework for component coupling that is developed as a part of the Center for Simulation of RF Wave Interactions with Magnetohydrodynamics (CSWIM) project (http://cswim.org/). The coupling between

different codes employs the IPS framework from CSWIM and the EFFIS framework from EPSI. In these studies it has been found that the stability of H-mode pedestal depends on non-local effects. This conclusion became possible because of the coupling of high fidelity codes together.

As individual physics components mature and are validated, they can be used in similar integrated studies. We anticipate a significant increase of the integrated modeling studies in the next five years. A key element in these studies will be the validation of the integrated modeling results. New techniques that facilitate such integration are being developed. Some of these techniques are based on the uncertainty quantification (UQ) analysis. Recently, the UQ DAKOTA toolkit has been coupled with the FACETS and TRANSP codes. The DAKOTA/FACETS simulations were used for the validation of transport models and the DAKOTA/TRANSP simulations were used for the quantification of uncertainties in the fusion Q prediction in the ITER tokamak. It has been demonstrated that the use of the UQ methods in the integrated modeling studies can be an extremely efficient tool for model validation, discharge optimization, and for predictive and interpretive analysis of the tokamak plasmas. However, these techniques significantly increase the use of computer resources.

The scientific objectives for the next several years will be the improvements of the coupling techniques and the development of the integrated modeling frameworks that will facilitate the coupling of plasma core, edge and wall models in tokamaks. The verification and validation of the integrated modeling results will be a top priority. The development of UQ techniques for the predictive studies and for the analysis of experimental data will continue. The use of UQ methods in the interpretive analysis is of a particular importance because it will help to standardize the computations of the error bars for the physics quantities that are derived from the raw experimental data.

Draft

12.1.1.3 Computational Strategies (now and in 2017)

The TRANSP code is a legacy integrated modeling code that has been developed as a serial code. There are some efforts on the parallelizing the components of the TRANSP code. In particular, the NUBEAM module for the neutral beam heating has been parallelized. The parallel full-wave TORIC code for ICRF auxiliary heating has been recently implemented in TRANSP.

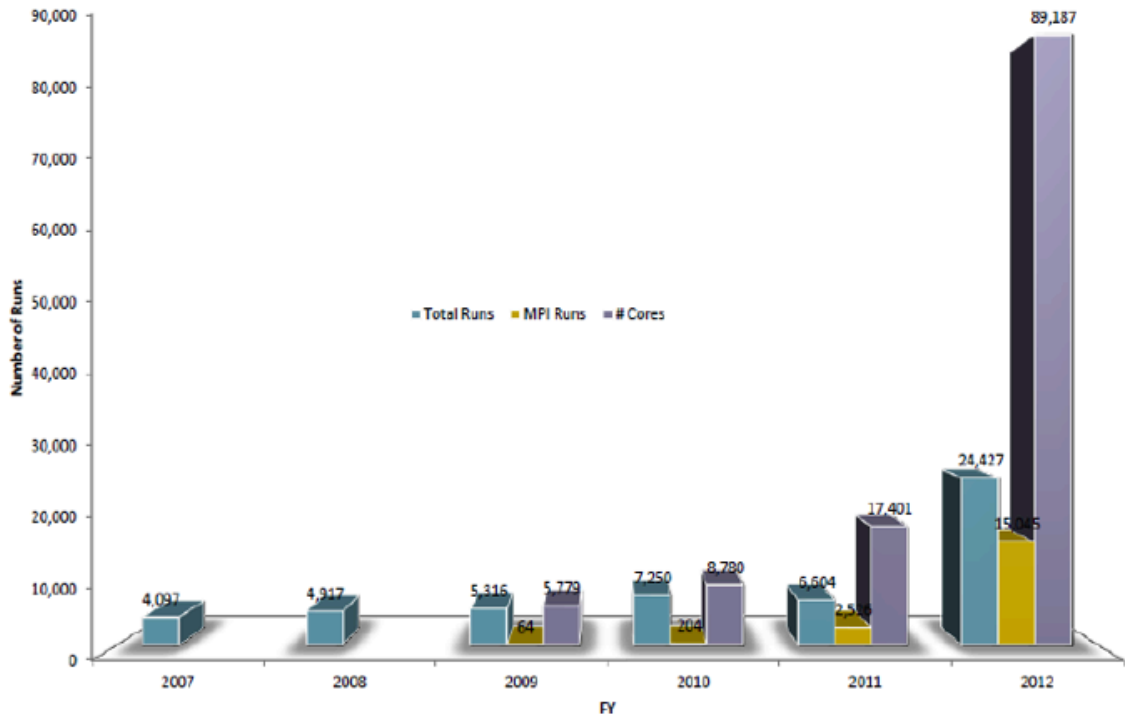


Figure 2. Number of TRANSP runs during the years of 2007-2012 [S. Jardin et al. TRANSP FY12 Usage Statistics and Overview, APS DPP meeting (Providence RI, October 2012)]. The total number of runs is shown as blue bars, the number of MPI runs is shown in yellow and number of cores used is shown in purple.

Finally, the parallel transport solver PT_SOLVER is being implemented and tested. The NUBEAM, TORIC, and PT_SOLVER modules utilize individual communication groups. The NUBEAM module was initially developed to run in serial as a component of the TRANSP code. The parallel version of NUBEAM demonstrates significant speed improvements. However, the particle advance per time step in NUBEAM is still 1000-10000 times slower comparing to modern particle codes. All parallel modules in TRANSP are implemented utilizing the original TRANSP framework that is not optimized for a parallel environment. As a result, a significant portion of run time is spent in an idle mode. Despite the obvious need for the framework and algorithmic improvements, the TRANSP code remains superior in terms of physics modeling capabilities. A wide selection of validated physics components makes the TRANSP code an attractive tool for the analysis of experimental data and for the predictive modeling of future tokamak discharges. The number of TRANSP simulations has continued to grow during the last 10 years as shown in Fig. 2. The majority of these simulations is for the analysis of the experimental results and MPI usage is typically associated with the NUBEAM and TORIC components. The approaching ITER commissioning will result in an increase in the demand for the predictive simulations. We expect that the number of TRANSP simulations will continue to grow in the next five years. However, we think that the increasing demand for predictive simulations will be restrained by the fundamental limitations in the computational framework and algorithmic deficiency.

The ITER size results additional requirements for the TRANSP components. In particular, a significantly larger number of Monte-Carlo particles in NUBEAM is needed than is needed for existing tokamaks. Significant code improvements are required for ITER simulations.

The FACETS code is a relatively new code that has been developed with a fresh start. Particular emphasis was given to the development of a parallel framework that facilitates the coupling of new and legacy physics modules and that allows the incorporation of modules for edge and plasma-wall interactions. An important aspect of this framework is the optimized load balancing that minimizes the idle time. Fig. 3 shows the distribution of CPU cores between different physics modules in the coupled core-edge transport simulation using the FACETS code. Similar to the PT_SOLVER in the TRANSP code, a new parallel transport solver in FACETS has been developed. The FACETS solver uses a Newton solve for the time advance with a multilevel method to enlarge the radius of convergence. As with a new code, the selection of physics module in FACETS is still somewhat limited. In addition, the FACETS code utilizes the same physics components as in the TRANSP code. These physics modules have the same algorithmic limitations as in TRANSP. The user base of the FACETS code is still very limited. However, we expect that it will grow as new physics components are implemented.

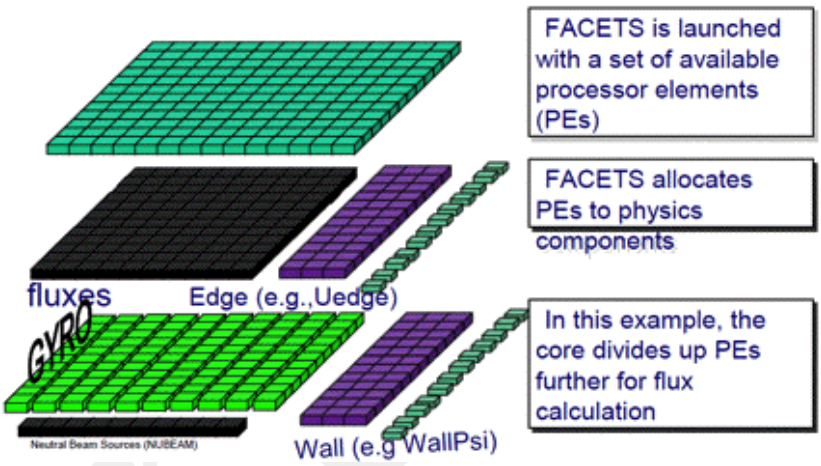


Figure 2. Use of CPU cores by different physics modules in the coupled core-edge transport simulation using the FACETS Code

In order to efficiently use the FACETS and TRANSP codes on HPC, additional code and algorithmic developments in both codes are necessary. The framework improvement in TRANSP and the extension of physics components in FACETS are needed. Some of these developments are already scheduled and expected to be completed by 2017.

12.1.2 HPC Requirements in 2017

There are three factors that will drive the number of integrated WDM simulations in the next four years. The first factor is related to an increasing demand on the interpretive analysis of experimental data. The commissioning the NSTX-U tokamak in 2014 and restarting the Alcator C-Mod operation will contribute to the number of interpretive TRANSP runs. The second factor is related to the approaching ITER commissioning. During the period prior to the production of the first ITER plasma, there will be an increasing market for the predictive simulations. Finally, the third factor is related to the implementation of novel techniques such as uncertainty quantification methods and design optimization techniques in the whole device modeling codes. These techniques can be used in the interpretive analysis of experimental data and for the validation and verification of physics modules in the transport codes. Because of these three driving factors, the total number of integrated WDM simulations is expected to exceed 60,000 in 2017. The number of CPU cores used in these simulations will increase at much rapid pace. The total number of

cores utilized in the integrated WDM simulations is expected to grow at least tenfold in 2017. Such significant increase in the number of cores can be explained by the development of physics components with improved fidelity that can be executed in parallel. In particular, the TGLF module for the anomalous transport has been implemented in the TRANSP code and the TGLF and GYRO modules can be executed from the FACETS framework. It is expected that transport core solvers in the TRANSP and FACETS codes will utilize up to 2000 and 20000 cores correspondingly. The auxiliary heating modules such as NUBEAM and TORIC are expected to utilize up to 1000 cores. Use of the HPC resources will be based on modules required for the appropriate physics conditions in a particular simulation. In order to optimize the load balancing and to accommodate long times in queues that are often required for the integrated modeling codes, a dedicated cluster for the integrated modeling at NERSC with ability to launch jobs on Hopper and other resources is needed.

12.1.2.1 Computational Hours Needed

Our estimate is based on the assumption that most simulations will be the interpretive simulations with the TRANSP code. The estimated number of interpretive TRANSP runs is 35,000 per year in 2017. The number of predictive simulations with the TRANSP and FACETS codes will continue to grow to 20,000 and 5,000, respectively. The interpretive simulations in TRANSP utilize a smaller number of cores than the predictive simulations. The average number of cores in the TRANSP runs is estimated as 24 for the interpretive simulations and 128 for the predictive simulations. Because of the option to use the gyro-kinetic GYRO code within the FACETS framework, the average number of cores in the predictive FACETS runs is expected to grow to 1,000 in 2017. Based on the average number of hours per run for the interpretive TRANSP, predictive TRANSP and predictive FACETS (6 hours, 50 hours, and 25 hours, respectively), we estimate the total number of core-hours that will be needed for the integrated Whole Device Modeling simulations in 2017 is 260M core-hours.

12.1.2.2 Parallelism

The number of cores needed for the interpretive TRANSP runs is somewhat limited. The only modules that utilize MPI resources in the interpretive mode are the modules for auxiliary heating such as NUBEAM and TORIC. Since there will be no need for the interpretive ITER runs in 2017, the requirements for the number of cores in NUBEAM and TORIC are not particularly stringent. Most interpretive simulations can be completed using from 16 to 32 cores. However, we anticipate some high-resolution simulations for which the number of cores will need to be increased to 256 or more. The TRANSP code does not currently support the full interpretive mode when the fluxes from predictive modules are directly compared with the fluxes found in the interpretive analysis. While there are plans to implement this option in TRANSP, the actual work is not yet started, and we cannot predict if this option will be available in 2017. The predictive TRANP runs can utilize up 2,000 cores and the FACETS runs can utilize up to 20,000 cores. Because of the improved load balancing, the FACETS code has relatively high scalability.

12.1.2.3 Data I/O

The size of data files produced in the integrated modeling simulations depends on many factors such as interpretive vs. predictive runs, and models used for sources and anomalous transport. The typical size of data files produced in integrated modeling simulations is expected to be in the range from 50 MB to 75 GB. The upper estimate is for FACETS simulations that utilize the GYRO model for the anomalous transport and the NUBEAM module for the neutral beam injection. While not all of these data files need to be stored

permanently, the estimate takes into account the restart capabilities that are available both in the TRANSP and FACETS codes. The FACETS code also supports MPI-IO.

Based on the number of predictive and interpretive runs the total storage space can be estimated as 200 TB. About one third of these data files will need to be archived on HPSS.

12.1.2.4 Memory Required

16GB per node is a reasonable memory requirement both for the TRANSP and FACETS simulations.

12.1.2.5 Many-Core and/or GPU Architectures

There are plans to port some reduced modules for anomalous transport and auxiliary heating. However, these projects are at very early stages and it is difficult to predict if any reasonable progress can be made to be implemented in the integrated modeling codes before 2017. However, the FACETS code can take an advantage of the existent GYRO development on porting the code to the GPU architecture.

12.1.2.6 Software Applications and Tools

We will continue to use the standard libraries at NERSC such as netCDF, HDF5, LAPACK, BLAS, SUPERLU, and PETSC.

Draft

Appendix A. Attendee Biographies

Application Scientists

Paul Bonoli received a PhD from Cornell University in 1981 under Professor Edward Ott. The title of his dissertation was "The Effects of Toroidal Geometry and Scattering by Density Fluctuations on the Accessibility and Energy Deposition of Lower Hybrid Waves". He is currently the Principal Investigator at MIT for the "Numerical Computation of Wave-Plasma Interactions SciDAC Project" funded by the Office of Fusion Energy Science and for the NSTX High Harmonic Heating and Current Drive Project at MIT. He is also now the co US Team Leader of the International Tokamak Physics Activity (ITPA) Working Group on Steady State Operation in ITER. Paul is a member of the American Physical Society and he is now serving as a member of the Executive Committee of the International Sherwood Controlled Fusion Theory Conference..

Jeff Candy is a principal scientist in the Fusion Theory Group at General Atomics, and lead developer of the GYRO code. His primary research foci are gyrokinetic and neoclassical theory/simulation, predictive transport modeling, and algorithms for parallel computing. Dr. Candy maintains a general but active interest in numerical methods, in particular those with application to plasma kinetic theory. Dr. Candy received his Ph.D. in 1994 under the supervision of M.N. Rosenbluth, was visiting scientist at the JET Joint Undertaking in the United Kingdom, and has been at General Atomics since 1998. He has received various Canadian awards, including 4 NSERC postgraduate fellowships, 2 NSERC postdoctoral scholarships, and the Sir James Lougheed Award of Distinction. In 2003, he was the inaugural recipient of the Rosenbluth award for fusion theory. He was 2008 Jubileum Professor at Chalmers University in Sweden, and was elected fellow of the American Physical Society in 2009. Dr. Candy is the author of approximately 80 refereed journal articles.

Choong-Seock (C-S) Chang is a Principal Research Physicist at the Princeton University Plasma Physics Laboratory (PPPL), and a Joint Professor of Physics at the Korea Advanced Institute of Science and Technology (KAIST). C.S. Chang is a Fellow of the American Physical Society. He serves in numerous national and international scientific committees, including the Council of the US Burning Plasma Organization (USBPO), Executive Committee for U.S. Transport Task Force (TTF), Theory Coordinating Committee (TCC) for DOE FES, Users' Council for National Center for Computational Sciences (NCCS), International Tokamak Physics Activity (ITPA), and others. He is the head of the Center for Edge Physics Simulation (EPSI) Scientific Discovery through Advanced Computing (SciDAC) project.

Stephane Ethier is a Computational Physicist in CPPG at the Princeton Plasma Physics Laboratory. Previously, he was a postdoctoral researcher in the Applied Physics group of the Mechanical and Aerospace Engineering Department of Princeton University, a computer consultant at INRS-Energie et Materiaux, and a research assistant at McGill University in Montreal, Canada. He received his Ph.D. from the Department of Energy and Materials Institut National de la Recherche Scientifique (INRS), Montreal, Canada. He has been the recipient of numerous awards including two postdoctoral fellowships from the Fonds pour la Formation de Chercheurs et Aide a la Recherche, the Lumonics Student Paper Competion Award, High Performance Computing Centre Award, National Sciences and the Engineering Research Council Award for Summer Research in both plasma physics and biophysics. He

has published several articles in refereed journals in addition to being a contributor to international conferences.

Alex Friedman received his B.S. and Ph.D. degrees (in Engineering Physics and Applied Physics, respectively) from Cornell University. He is a physicist with Lawrence Livermore National Laboratory's Fusion Energy Sciences Program, and an Affiliate at Lawrence Berkeley National Laboratory. He is a Fellow of the American Physical Society, a recipient of the LLNL Physics Department's Distinguished Achievement Award, and a past Associate Editor of the Journal of Computational Physics. His research interests include computational plasma and particle beam physics, computational electromagnetics, accelerator physics and engineering, methods for data analysis, and numerical analysis. He has authored over 100 published papers and numerous reports.

Kai Germaschewski is Assistant Professor in the Space Science Center of the Institute for the Study of Earth, Oceans and Space at University of New Hampshire. he received his Ph.D. in Computational Plasma Physics from the University of Duesseldorf, Germany in 2001. His advisor was Rainer Grauer and his thesis is titled "Pulse propagation in media with anisotropic dispersion". After working one year as a post-doc at the Ruhr-University Bochum, Germany he joined Amitava Bhattacharjee's Center for Magnetic Reconnection Studies (CMRS) at the University of Iowa in 2002, working on Hall-MHD simulation codes employing Adaptive Mesh Refinement. He moved together with the group to the University of New Hampshire in summer 2003. His research aims to gain a better understanding of fast reconnection processes in two-fluid systems, applicable to laboratory as well as space plasmas. The focus of his work is on sophisticated, high performance, massively parallel numerical methods, in particular block-structured adaptive mesh refinement to efficiently resolve a large range of spatial scales and implicit Newton-Krylov-Schwarz based methods to overcome stability limitations present in explicit numerical schemes.

Stephen Jardin is the Theory Department Facilitator for MHD. He is also co-head of the Computational Plasma Physics Group and head of Physics within the Next Step Option design effort. Jardin, a Principal Research Physicist, has been on the Plasma Physics Faculty of Princeton University with rank of Professor since 1986. He currently teaches a graduate course in Computational Methods in Plasma Physics. He has been a member of the National Energy Research Supercomputer User Group Executive Committee since 1992 and became Chairperson of its Program Advisory Committee in 1999. He is also a member of the ESnet Steering Committee and Chairman of the National Transport Code Collaboration Program Advisory Committee. He is a Fellow of the American Physical Society. Jardin is the author of more than 150 refereed papers, holds four U.S. patents, and has played a key role in the development of several large magnetohydromagnetic (MHD) computer programs now widely used in magnetic fusion research. He led the MHD design effort for the Burning Plasma Experiment (BPX) and Tokamak Physics Experiment (TPX) experimental proposals. He is the leader of the physics unit of the ARIES studies. Jardin received a bachelor's degree in engineering physics with highest honors from the University of California, Berkeley, in 1970, where he was elected a member of the Phi Beta Kappa. After receiving a National Science Foundation graduate fellowship, he received a master's in physics and a master's in nuclear engineering from the Massachusetts Institute of Technology in 1973. In 1976, he received a Ph.D. in Astrophysical Sciences, Plasma Physics Section, from Princeton University.

Homa Karimabadi is the founder of SciberQuest and co-founder of three other startups including Aitria Technologies. He has a unique combination of expertise in supercomputing

techniques and information technology as well as strategic analysis and business management. As a scientist, he is recognized for key discoveries in several fields and acted as a consultant for SAIC. He has published over 70 articles in scientific journals on a wide range of topics including cosmology, chaos theory, space plasmas, numerical algorithms, and solar physics. On the business side, he guided the strategic planning and business development for a venture funded internet start-up, including formulation/presentation of business models. He also worked as a lead strategist for an Internet Incubator founded by a McKinsey Partner. In addition to his role as the CEO/CTO at SciberQuest, Inc., Dr. Karimabadi is also heading the space physics plasma simulation group at UCSD. He received his B.S. in mathematics and astronomy from UC Berkeley and his PhD in plasma astrophysics from the University of Maryland.

Alice Koniges became the first woman ever to earn a PhD in Applied and Computational Mathematics at Princeton University. She began her career as a member of NERSC's Computational Physics Group in 1984 at LLNL. She achieved the first successful parallel code run on the four-processor Cray-2. She began her career researching parallel computing and plasma physics, focusing on the combination of these two fields. Her expertise in the transition from vector to parallel computing culminated in her textbook Industrial Strength Parallel Computing, published by Morgan Kaufmann Publishers in January 2000. Alice joined the Berkeley Lab in 2009. Her current research interests include programming models for multicore architectures, benchmarking and performance optimization of application codes, development of Adaptive Mesh Refinement (AMR) and Arbitrary Lagrangian Eulerian (ALE) algorithms for time-dependent PDE's, and application supercomputing in plasma physics, laser physics, and energy research. She regularly gives tutorials and short courses on application supercomputing. She also manages the NERSC Petascale post-doc program as Principal Investigator of the Computational Science and Engineering Petascale Initiative at LBNL. Previous to joining the Berkeley Lab, she held various positions at the Lawrence Livermore National Laboratory, including management of the Lab's institutional computing. She also led the effort to develop a new 3D multiscale multiphysics code (ALE-AMR) that is used to predict the impacts of target shrapnel and debris on the operation of the National Ignition Facility (NIF) the world's most powerful laser, and model Warm Dense Matter (WDM) experiments at the NDCX facility at LBNL. From 1995 to 1997, Alice led the Parallel Applications Technology Program at LLNL. This was the LLNL portion of the largest (12 million) CRADA (Cooperative Research and Development Agreement) ever undertaken by the Department of Energy. She spent 1998 at the Max-Planck Institute in Garching, Germany (Computer Center and Plasma Physics Institute), where she was a consultant to users at the Institute, assisting in the conversion of applications codes for parallel computers. In addition to her PhD she also holds MSE and MA degrees from Princeton, and a BA from the University of California, San Diego and has published approximately 100 refereed technical papers.

Zhihong Lin is a Professor of Physics & Astronomy in the School of Physical Sciences at University of California, Irvine. In 2000, he received the Presidential Early Career Award for Scientists and Engineers. He was also the recipient of the Kaul Foundation Prize for Excellence in Plasma Physics research and Technology Development in 1999 for performing advanced simulations with unprecedented realism and resolution leading to results demonstrating the positive impact of modern massively parallel computers and for outstanding contributions to understanding the physics of sheared zonal flows. Prior to his current position, he was a staff research physicist in the Theory Department at PPPL and a DOE Fusion Energy Postdoctoral Fellow. Dr. Lin received his Ph.D. in Plasma Physics from

Princeton University - Department of Astrophysical Sciences - Program in Plasma Physics in 1996. He received a B.S. in physics from Beijing University (China) in 1989.

Vyacheslav (Slava) Lukin holds the position of an Astrophysicist in the Space Science Division of the US Naval Research Laboratory and is an Affiliate Assistant Professor of Aeronautics and Astronautics at the University of Washington. He received B.A. (2000) in Physics and Mathematics from Swarthmore College, and M.A. (2003) and Ph.D. (2008) in Astrophysical Sciences from Princeton University. Dr. Lukin's research interests span the gamut from the physics of solar and space plasmas, to space weather prediction, to numerical simulations of laboratory plasma experiments, to development of scalable high-order methods for modeling the dynamics of highly nonlinear systems, to uncertainty quantification in and validation of simulations of such systems. He is the lead principle developer of the high-order finite element multi-fluid modeling framework HiFi. Dr. Lukin collaborates with students and researchers utilizing the HiFi framework to study plasmas in a variety of domains and parameter regimes: in basic laboratory plasma experiments, e.g., the Reconnection Scaling Experiment (LANL) and the Swarthmore Spheromak Experiment (Swarthmore College), in hot fusion experiments, e.g., the Mega Ampere Spherical Tokamak (Culham Centre for Fusion Energy, United Kingdom), and for modeling the Earth's ionospheric, magnetospheric and solar plasmas in collaboration with research groups at the University of Wisconsin, LANL, Harvard-Smithsonian Center for Astrophysics, NASA Goddard, and the University of Manchester.

John Mandrekas is a Program Manager at the DOE Office of Science's Fusion Energy Sciences program, where he manages the theory and advanced simulation programs and the High Performance Computing resources. Previously, he was a Senior Research Scientist at Georgia Tech in Atlanta, Georgia, where he did research in theoretical and computational fusion plasma science and taught undergraduate and graduate level courses in plasma physics and fusion. He received a Ph.D. in Nuclear Engineering from the University of Illinois at Urbana-Champaign in 1987.

Alexei Pankin holds a Ph.D. in Physics from Institute for Nuclear Research (1999). His dissertation was, "Turbulence of Non-Equilibrium Plasma Near Boundary of Marginal Stability," (Thesis Advisor: Prof. Tatiana A. Davydova). His research interests are dynamical systems, plasma physics, fluid dynamics, and numerical analysis, particularly: Subcritical transition to turbulence in plasmas; subcritical turbulence; Waves and instabilities; Diffusion processes in multicomponent gas/plasma mixtures; Numerical simulation of processes in devices of plasma deposition; Self-organized critically (SOC) systems; Non-linear optics: soliton propagation in optical fibers, stimulated backscattering; guided wave optics;; Neural networks; and Linear stability analysis.

Scott Parker is Professor of Physics at University of Colorado. His research is in the area of kinetic theory and simulation of plasmas. Most of my current work is in the area of direct numerical simulation of tokamak plasma turbulence on large massively parallel computers. These simulations are done in a five dimensional phase space using newly developed particle-based methods. These calculations involve many millions of particles with self-consistently calculated electric fields. For the first time, these simulations are showing spectral features and transport levels at least qualitatively similar to turbulent transport observed in large present day experiments. Our goal is a fundamental understanding of plasma turbulence and transport in magnetized plasmas. Scientific visualization is utilized to analyze the features of the three dimensional turbulent fluctuations.

Chuang Ren is Assistant Professor of Mechanical Engineering and Physics at University of Rochester. Prof. Ren received his B.S. (1990) and M. S. (1993) degrees in Physics from Tsinghua University, Beijing, China. He received his Ph.D. in Physics from the University of Wisconsin-Madison in 1998. He was a postdoctoral researcher and then a staff researcher at UCLA before joining the University as an Assistant Professor of Mechanical Engineering and a Scientist of the Laboratory for Laser Energetics in 2004. In 2005, he received a joint appointment as Assistant Professor of Physics. In 2006, he was awarded a Faculty Development Award in Plasma Physics from the Department of Energy. Prof. Ren's research interest is generally in the area of theoretical and computational Plasma Physics. In particular, he works to apply plasma physics to a wide range of applications including astrophysics, inertial confinement fusion (ICF), plasma-based accelerators, and new radiation sources.

Carl R. Sovinec is a Professor of Engineering Physics at the University of Wisconsin-Madison. Professor Sovinec's research interests lie in the numerical simulation of plasmas and fluids. His recent efforts have focused on simulating nonlinear electromagnetic behavior in magnetically confined plasmas, where the extreme stiffness and anisotropy resulting from the magnetic field provide great challenges for numerical approaches. The work is aimed toward providing comprehensive modeling of magnetically confined plasmas. Such modeling will advance the effort of achieving controlled fusion reactions for energy production by helping us understand dynamics observed in existing devices and by allowing us to optimize new configurations before hardware is constructed. As an academic, Professor Sovinec hopes to share his enthusiasm for numerical simulation with new generations of physicists and engineers. Professor Sovinec has active collaborations with a number of organizations across the country and worldwide. As a major contributor to the NIMROD (Non-Ideal Magnetohydrodynamics with Rotation) code development team, http://nimrodteam.org, he regularly collaborates with team members at Tech-X Corporation, Utah State University, and the University of Colorado-Boulder. He also consults with NIMROD code users at Lawrence Livermore National Laboratory and the University of Washington.

Linda Sugiyama is a member of the Research Laboratory of Electronics (RLE) at the Massachusetts Institute of Technology High Energy Plasma Physics Group. She received a BS in applied mathematics from the University of Wisconsin, Madison in 1975 and a PhD in mathematics from MIT in 1980. She became a postdoctoral associate in RLE in 1980, joining the RLE research staff in 1983. Sugiyama's research has centered on a continuing interest in many-body interacting systems, with a focus on the physics of plasmas in magnetic fields and on the development of magnetically confined plasmas for thermonuclear fusion. Because of the complexity of these many-body systems, a great deal of Sugiyama's work has been directed toward advancing and using computational models for simulating their behavior. A major recent direction for her research has been efforts to simulate a confined plasma's time evolution of a two-fluid magnetohydrodynamic model, in which the electronics and ions are treated separately, rather than the more conventional single-fluid approach. Sugiyama, widely known and respected in the field of plasma physics, is also active in professional organizations, such as the American Physical Society for which she serves as a member of the Committee on Women in Physics.

Frank S. Tsung is a researcher in the Department of Physics & Astronomy at the University of California at Los Angeles. His research interests include particle-in-cell simulations of wave particle interactions, including laser wakefield accelerators, laser plasma interactions in inertial confinement fusion plasmas, and wave particle interactions in space and

laboratory plasmas. He received his B.S. from UC Berkeley and his Ph.D. from UCLA under the supervision of John M. Dawson. He is a principal architect is one of the code OSIRIS and is a current member of NUGEX, the NERSC User Group's Executive Committee.

Brian D. Wirth joined the University of Tennessee Nuclear Engineering Department in July 2010 as the ninth University of Tennessee-Oak Ridge National Laboratory Governor's Chair. Wirth was previously an associate professor at the University of California, Berkeley, which he joined in 2002 following several years as a materials scientist at Lawrence Livermore National Laboratory. Wirth leads a number of research projects funded by various U.S. Department of Energy offices to investigate the performance of nuclear fuels and structural materials in nuclear environments. The research is planned to lead to improved predictions of the longevity of nuclear reactor components and ultimately the development of high-performance, radiation resistant materials for advanced nuclear fission and fusion energy applications. Wirth received a B.S. in Nuclear Engineering from the Georgia Institute of Technology in 1992 and a Ph.D. in Mechanical Engineering from the University of California, Santa Barbara in 1998, where he was a Department of Energy Nuclear Engineering Graduate Fellow.

Xueqiao Xu is a theoretical and computational plasma physicist with general interests in plasma physics, fluid mechanics, statistical mechanics, and parallel computing. Specific plasma physics experience: particle simulation, continuum simulation, fluid simulation, linear and nonlinear theory of micro-instabilities, turbulence, transport, and boundary plasma physics. Specific achievements include: (1) Developed the BOUT two-fluid edge turbulence code; (2) Lead physics developer for 5D continuum edge gyrokinetic code TEMPEST; (3) Co-invented new Monte Carlo scheme for usual gyro-kinetic δf particle-in-cell plasma micro-turbulence simulations for modeling Coulomb collisions. Over 60 refereed publications.

Editors and NERSC Application Support Personnel

Richard Gerber is NERSC Senior Science Advisor and User Services Deputy Group Lead and, with Harvey Wasserman, organizes the NERSC High Performance Computing and Storage Requirements Reviews for Science and edits the reports. He holds a Ph.D. in physics from the University of Illinois at Urbana-Champaign, specializing in computational astrophysics; held a National Research Council postdoctoral fellowship at NASA-Ames Research Center 1993-1996; and has been on staff at NERSC since.

Harvey Wasserman is a member of the NERSC User Services Group and helps to organize the NERSC High Performance Computing and Storage Requirements Reviews.

Appendix B. Workshop Agenda

Tuesday, March 19		
Time	Topic	Presenter
8:00am	Arrive, informal discussions	
8:30 AM	Welcome, Overview of Requirements Reviews	Dave Goodwin, ASCR (NERSC Program Manager)
8:45 AM	Future Directions in Fusion Energy Science Research	John Mandrekas, FES Team Leader, Theory & Simulation
9:15 AM	Workshop Goals, Process	Richard Gerber, NERSC User Services Deputy Group Leader and NERSC Senior Science Advisor
	Magnetic Fusion Energy Sciences (MFES) Case Studies	
10:00 AM	Core Turbulence and Transport	Jeff Candy, General Atomics
10:30 AM	Energetic Particles	Zhihong Lin, U.C. Irvine
11:00 AM	RF / Plasma Interactions	Paul Bonoli, MIT
11:30 AM	Plasma Edge Physics	C.S. Chang, PPPL
12:00 PM	Group Photo	
12:30 PM	Working Lunch Presentation: NERSC's 10-Year Plan	Sudip Dosanjh, NERSC Director
	Magnetic Fusion Energy Sciences (MFES) Case Studies (continued)	
1:30 PM	Macroscopic Stability	Stephen Jardin, PPPL; Carl Sovinec, U. Wisconsin
2:00 PM	Small Scale Experimental Plasma Research	Vyacheslav Lukin, NRL
	General Plasma Science Case Study	
2:30 PM	General Plasma Science	Kai Germaschewski, U. New Hampshire; Homa Karimabadi, UC San Diego
3:00 PM	PM Break	
	High Energy Density Laboratory Plasmas (HEDLP) & Inertial Fusion Energy Sciences (IFES) Case Studies	
3:15 PM	Heavy Ion Beams	Alex Friedman, LLNL
3:45 PM	Laser Plasma Interactions	Chuang Ren, University of Rochester
4:15 PM	PM Break	
	Materials and Plasma-Surface Interactions Case Study	
4:30 PM	Materials and Plasma-Surface Interactions	Brian Wirth, UTK
6:00 PM	Adjourn	

Wednesday, March 20		
8:00 AM	Informal discussions	
8:30 AM	Data Discussion	
9:15 AM	Integrated Modeling	Alexei Pankin, TechX, and Arnold Kritz, Lehigh U.
10:00 AM	AM Break	
10:15 AM	Day 1 Summary	Richard Gerber & Harvey

Draft

		Wasserman
10:45 AM	High-level findings	
11:15 AM	Schedule for Report	Harvey Wasserman
12:00 PM	Working Lunch: Case Study Breakout Sessions	All
1:00 PM	Adjourn	

Draft

Appendix C. Abbreviations and Acronyms

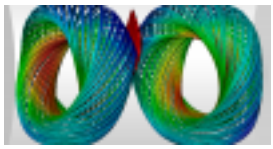
ALCC	ASCR Leadership Computing Challenge
ALCF	Argonne Leadership Computing Facility
AMR	Adaptive Mesh Refinement
ASCR	Advanced Scientific Computing Research, DOE Office of
AY	Allocation Year
BER	Biological and Environmental Research, DOE Office of
CUDA	Compute Unified Device Architecture
ELM	Edge localized mode
ESG	Earth System Grid
ESnet	DOE's Energy Sciences Network
FEM	Finite Element Modeling
FFT	Fast Fourier Transform
GPGPU	General Purpose Graphical Processing Unit
GPS	General plasma science
GPU	Graphical Processing Unit
HDF	Hierarchical Data Format
HEDLP	High Energy Density Laser Plasma
HPC	High-Performance Computing
HPSS	High Performance Storage System
I/O	input output
IDL	Interactive Data Language visualization software
IFES	Inertial Fusion Energy Science
INCITE	Innovative and Novel Computational Impact on Theory and Experiment
LANL	Los Alamos National Laboratory
LBNL	Lawrence Berkeley National Laboratory
LLNL	Lawrence Livermore National Laboratory
MFES	Magnetic Fusion Energy Science
MHD	Magnetohydrodynamics
MPI	Message Passing Interface
NASA	National Aeronautics and Space Administration
NERSC	National Energy Research Scientific Computing Center
NetCDF	Network Common Data Format
NGF	NERSC Global Filesystem
NISE	NERSC Initiative for Science Exploration
NRL	Naval Research Laboratory
OLCF	Oak Ridge Leadership Computing Facility
ORNL	Oak Ridge National Laboratory
OS	o operating system
PDE	Partial Differential Equation
PDSF	NERSC's Parallel Distributed Systems Facility
PPPL	Princeton Plasma Physics Laboratory
SC	DOE's Office of Science
SciDAC	Scientific Discovery through Advanced Computing
SLAC	SLAC National Accelerator Laboratory
UQ	Uncertainty Quantification

Draft

Appendix D. About the Cover

Image showing a portion of NERSC's "Hopper" system, a Cray XE6 installed during 2010. Hopper is NERSC's first peta-FLOP resource, with a peak performance of 1.28 PetaFLOPs/sec, 153,216 compute cores, 212 Terabytes of memory, and 2 Petabytes of disk. Hopper placed number five on the November 2010 Top500 Supercomputer list.

Evolution of electrical current density, parallel to magnetic field, in the Pegasus Toroidal Experiment at the University of Wisconsin-Madison, as simulated by graduate student John O'Bryan with the NIMROD 3D magnetohydrodynamics code. Image courtesy of Prof. Carl Sovinec, University of Wisconsin



HiFi simulation of reconnection and merging of two spheromaks in the Swarthmore Spheromak Experiment showing representative magnetic field lines. Peak in current density is represented by the surface in the center of the volume. From Gray et al., Physics of Plasmas 17, 102106 (2010). Image courtesy of Vyacheslav Lukin, Naval Research Laboratory.

Draft