Application Software Case Studies in FY05 for the Mathematical, Information and Computational Sciences Office of the U.S. Department of Energy

March 7, 2006

Point of contact: K. J. Roche[1] , rochekj@ornl.gov

AORSA: E. F. Jaeger[2], L.A. Berry[2], D. B. Batchelor[2], E. DAzevedo[3], M. D. Carter[2], R. Barrett[1], R. W. Harvey[4], R .J. Dumont[5], J. R. Myra[6], D. A. DIppolito[6],D. N. Smithe[7], C. K. Phillips[8],P. T. Bonoli[9], J. C. Wright[9]

CCSM: J. Drake[10], P. Worley[10],P. Jones[11]

LAMMPS: S. J. Plimpton[12]

Omega3P: K. Ko[13], N. Folwell[13], L. Ge[13], L. Lee[13], Z. Li[13], C. Ng[13], G. Schussman[13], R. Uplenchwar[13], L. Xiao[13], E. Ng[14], W. Gao[14], P. Husbands[14], X. Li[14], C. Yang[14], G. Golub[15], K. Ma[16], Z. Bai[16], L. Diachin[17], D. Brown[17], P. Knupp[18], T. Tautges[19], M. Shephard[20], A. Brewer[21], E. Seol[21], D. Keyes[22], O. Ghattas[23], V. Akcelik[23]

S3D INCITE: J. H. Chen[24], E. R. Hawkes[24], R. Sankaran[24], J. H. Sutherland[24], M.R. Fahey[29], D. Skinner[25]

S3D SciDAC: H. G. Im[26], C. S. Yoo[26], C. J. Rutland[27], Y. Wang[27],

A. Trouve[28], Y. Wang[28]

[1]Computer Science and Mathematics Division, Future Technologies Group, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6173 Oak Ridge, TN 37831-6008, U.S.A

[2]Fusion Energy Division, Plasma Theory Group, Oak Ridge National Laboratory,One Bethel Valley Road, P.O. Box 2008, MS - 6169 Oak Ridge, TN 37831-8071, U.S.A

[3]Computer Science and Mathematics Division, Numerical Mathematics, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6016 Oak Ridge, TN 37831-6016, U.S.A

[4]CompX, San Diego, CA

[5]Association EURATOM-CEA sur la Fusion, CEA/DSM/DRFC

[6]Lodestar Research Corporation, 2400 Central Avenue P-5, Boulder, Colorado 80301

[7]ATK-Mission Research, 8560 Cinderbed Rd., Suite 700, Newington, VA 22122-8560

[8]Princeton Plasma Physics Laboratory, Princeton, New Jersey 08543

[9]Plasma Fusion Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

[10]Computer Science and Mathematics Division, Climate Dynamics Group, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6016 Oak Ridge, TN 37831-6016, U.S.A

[11]T-3, Los Alamos National Laboratory, MS B216, Los Alamos, NM 87545, U.S.A

[12]Sandia National Laboratories, Computational Biology and Evolutionary Computing, MS 1110, Albuquerque, NM 87185-1110 , U.S.A

[13]Stanford Linear Accelerator Center, MS 27, 2575 Sand Hill Road, Menlo Park, CA 94025, U.S.A

[14]Lawrence Berkeley National Lab, 1 Cyclotron Road Mail Stop 943, Berkeley, CA 94720-0252, U.S.A

[15]Stanford University, Department of Computer Science, Gates Building 2B, Room 280, Stanford, CA 94305-9025, U.S.A

[16]University of California-Davis, Department of Computer Science, 2063 Kemper Hall, One Shields Avenue, Davis, CA 95616-8562, U.S.A

[17]Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, U.S.A

[18]Sandia National Laboratories, Parallel Computing Sciences Department, Albuquerque, New Mexico, U.S.A

[19]Sandia National Laboratories, Computational Mechanics and Visualization, Livermore, CA 94551, U.S.A

[20]Rensselaer Polytechnic Institute, Department of Civil Engineering, Troy, NY 12180-3590, U.S.A

[21]Rensselaer Polytechnic Institute, Department of Computer Science, Troy, NY 12180-3590, U.S.A

[22]Columbia University, Applied Physics & Applied Mathematics, 200 S. W. Mudd Bldg., MC 4701, 500 W. 120th Street , New York, NY, 10027, U.S.A

[23]Carnegie Mellon University, Civil & Environmental Engineering, Pittsburgh, Pennsylvania 15213, U.S.A

[24]Sandia National Laboratories, Combustion Research Facility, Livermore, CA 94551, U.S.A

[25]National Energy Research Scientific Computing Center, Lawrence Berkeley National Lab, 1 Cyclotron Road Mail Stop 943, Berkeley, CA 94720-0252, U.S.A

[26]University of Michigan, Department of Mechanical Engineering, Ann Arbor, MI 48109

[27]University of Wisconsin-Madison, Department of Mechanical Engineering, Madison, WI 53706

[28]University of Maryland, Department of Fire Protection Engineering, College Park, MD 20742

[29]National Center for Computational Sciences, Scientific Computing Group, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6008, Oak Ridge, TN 37831-6008, U.S.A

Contents

1 Overview **13**
1.1 Determining software effictiveness . . . 14
1.2 Results . . . 15
1.2.1 S3D INCITE . . . 15
1.2.2 AORSA . . . 16
1.2.3 CCSM . . . 16
1.2.4 Omega3P . . . 17
1.2.5 LAMMPS . . . 17
1.2.6 S3D SciDAC . . . 18
1.3 Support Information . . . 19

2 S3D : Innovative and Novel Computational Impact on Theory and Experiment (INCITE) **19**
2.1 Introduction . . . 20
2.2 Background and Motivation . . . 20
2.3 Research Goals . . . 22
2.3.1 Extinction and Reignition Dynamics and Statistics . . . 22
2.3.2 Differential Diffusion . . . 23
2.3.3 A-priori Model Development and Validation . . . 23
2.3.4 DNS Benchmark for comparison . . . 24
2.3.5 LES Connection . . . 25
2.3.6 Experimental Connection . . . 25
2.4 Computational Approach . . . 25
2.5 Physical Configuration . . . 26
2.6 Physical and Numerical Parameter Selection Considerations . . . 27
2.7 Formulation . . . 30
2.7.1 Governing Equations . . . 31
2.7.2 Constitutive Relationships . . . 32
2.7.3 Stress Tensor . . . 33
2.7.4 Mass Diffusion Flux . . . 33
2.7.5 Heat Flux . . . 34
2.8 Code Overview . . . 34
2.9 S3D Software Performance and Improvements . . . 36
2.10 Test problem description . . . 37
2.11 Scalar profiling and performance improvements . . . 37
2.11.1 Performance on Seaborg . . . 37
2.11.2 Porting the code to Cray X1 . . . 39
2.12 Evaluation of parallel scaling and communication overheads . . . 41
2.12.1 Scaling on Seaborg . . . 41
2.12.2 Scaling on Cray X1 . . . 42
2.13 Evaluation of memory limitations . . . 43
2.14 Q3 Progress and Dynamics . . . 44
2.14.1 Performance improvements on Seaborg . . . 44
2.14.2 Performance improvements on Cray X1 . . . 45

2.15 Scientific Insights from DNS Simulations 46
2.15.1 Mixing timescales 46
2.15.2 Mathematical definition 46
2.15.3 Effect of Schmidt Number 47
2.15.4 Effect of Reaction-Diffusion coupling 48
2.16 Knowledge discovery from terascale simulated combustion data . 49
2.17 Prospects for future combustion simulations on petascale platforms - breakthrough science and what is required 50
2.18 Code Improvements made in FY05-Q4 54
2.18.1 Performance improvements on Cray X1E 54
2.18.2 Porting the code to XT3 55
2.18.3 Summary of Joule target accomplishment 56

3 AORSA 57
3.1 Background 57
3.2 Physics Basis for AORSA 59
3.2.1 Plasma conductivity tensor for non-Maxwellian components 60
3.2.2 Energy absorption by non-Maxwellian plasmas 62
3.2.3 Quasi-linear diffusion coefficients 64
3.3 Numerical Basis for AORSA 66
3.3.1 Code Description 66
3.3.2 Numerical solution of the wave equation 67
3.3.3 Loading the solution matrix and calculating the non-Maxwellian plasma conductivity tensor 69
3.3.4 Re-calulating the plasma current for each species 69
3.3.5 Calculating the non-Maxwellian power absorption and quasi-linear operator 69
3.3.6 Validating the numerical calculations 70
3.3.7 Choosing the hardware allocation for AORSA 70
3.4 Science Case: Non-Maxwellian bulk ion component in the NSTX tokamak 72
3.4.1 Case Study 72
3.4.2 Parameters 72
3.4.3 Observables 73
3.5 Science Case: Fusion born alpha particles in the ITER burning plasma 73
3.5.1 Case Study 73
3.5.2 Parameters 74
3.5.3 Observables 74
3.6 Q2 Recent Numerical Progress 75
3.6.1 Energy absorption in 1-D for non-Maxwellian plasmas . . 75
3.6.2 Energy absorption in 2-D for Maxwellian plasmas 75
3.6.3 Energy absorption in 2-D for non-Maxwellian plasmas . . 75
3.7 Q3 Progress and Dynamics 76
3.7.1 Physics Progress 76
3.7.2 Numerical Progress 76

3.8 Q4 Progress and Dynamics . . . 77
3.8.1 Physics Progress . . . 77
3.8.2 Numerical Progress . . . 78
3.9 Publications for the current fiscal year . . . 79

4 Community Climate System Model -CCSM 79
4.1 Overview of the SciDAC CCSM Software Development Project . 80
4.2 Algorithmic Basis . . . 81
4.3 CCSM Science Cases . . . 83
4.4 Case Study: Baseline Community Atmospheric Model . . . 84
4.4.1 Parameters . . . 84
4.4.2 Observables . . . 85
4.5 Case Study: The coupled Community Climate System Model . . 85
4.5.1 Parameters . . . 86
4.5.2 Observables . . . 87
4.6 Case Study: The Parallel Ocean Program . . . 87

5 Omega3P 87
5.1 Introduction . . . 87
5.1.1 Parallel Electromagnetic Codes developed under SciDAC 89
5.1.2 Validation of *Omega3P* . . . 89
5.2 Physical Basis for ***Omega3P*** . . . 91
5.2.1 Lossless Closed Cavity . . . 91
5.2.2 Closed Cavity with Lossy Materials . . . 92
5.2.3 Open Cavity with External Coupling . . . 92
5.3 Numerical Basis for *Omega3P* . . . 93
5.3.1 Finite Element Formulation of Eigenvalue Problem . . . 93
5.3.2 Methods of Solution . . . 94
5.3.3 Omega3P Code Description . . . 95
5.3.4 Choosing the Hardware Platform for *Omega3P* . . . 96
5.4 *Omega3P* Science Cases . . . 96
5.4.1 Omega3P Input Parameters . . . 97
5.4.2 Omega3p Output Observables . . . 97
5.5 Q1: TTF Cavity for the International Linear Collider . . . 98
5.5.1 Physics Output . . . 98
5.5.2 Algorithmic Implementation . . . 99
5.5.3 Computational Performance . . . 100
5.6 Q2: Low-loss Cavity for the International Linear Collider . . . 100
5.6.1 Physics Output . . . 100
5.6.2 Algorithmic Implementation . . . 100
5.6.3 Computational Performance . . . 101
5.7 Q3 Progress and Dynamics . . . 101
5.8 Q4 Progress and Dynamics . . . 103
5.8.1 Super Structure (SST) for the International Linear Collider 103
5.8.2 Physics Output . . . 103
5.8.3 Algorithmic Implementation . . . 103

5.8.4 Computational Performance 105

6 LAMMPS -a classical molecular dynamics package 105
6.1 Overview of Molecular Dynamics 105
6.2 Overview of LAMMPS 106
6.3 Code Development Plans for FY05 108
6.3.1 Algorithmic Improvements 108
6.3.2 Speed Enhancements 108
6.3.3 New Features 108
6.4 Science Plans for FY05 109
6.4.1 RuBisCO protein 110
6.4.2 Metal Islands on Silica 110
6.4.3 Slow Granular Flows 111
6.5 Benchmarking Effort for the Joule Project 113
6.6 Q1 Progress and Dynamics 116
6.7 Q2 Progress and Dynamics 116
6.8 Q3 Progress and Dynamics 117
6.9 Q4 Progress and Dynamics 117

7 S3D SciDAC 118
7.1 Objectives 118
7.2 Code Description 119
7.3 Overview of the TSTC Project 120
7.4 Science Case: Ethylene-Air Counterflow Nonpremixed Flames with Soot and Radiation 121
7.4.1 Case Study 123
7.4.2 Parameters 124
7.4.3 Results 124
7.4.4 Flame-vortex interaction 124
7.4.5 Summary of counterflow nonpremixed flame study 129
7.5 Science Case: Fuel Spray Injection and Mixing 129
7.5.1 Transport Equations 130
7.5.2 Case Study 132
7.5.3 Parameters 132
7.5.4 Code Observables 133
7.5.5 Physical Observables 133
7.5.6 Results 133
7.5.7 Summary of fuel injection study 137
7.6 Hardware and Software 137
7.7 Q3 Progress and Dynamics 138
7.7.1 Comparison of DTM and DOM in nonpremixed flames with soot formation 138
7.7.2 Effects of different formulation for the convective terms in the Navier-Stokes equations 139
7.8 Relation to Other DOE Initiatives 141
7.9 Q4 Progress and Dynamics 141

7.9.1 Accuracy improvement with DOM 142
7.9.2 Transient soot dynamics 142
7.9.3 Improved boundary conditions for spray jet simulations . 146
7.10 Summary of enhancements made in FY2005 146
7.10.1 Performance enhancements 146
7.10.2 Scientific enhancements 148

8 FY04 Reprint: Community Climate System Model 149
8.1 Overview of the SciDAC CCSM Software Project 149
8.2 CCSM Software Development Path 151
8.3 Performance Metrics 153
8.4 Hardware Bottlenecks 153
8.5 Case Study: The Community Atmospheric Model 153
8.5.1 Scientific Goals and Preparation for the IPCC Simulation Project 153
8.5.2 Resolution for Regional Impacts 154
8.5.3 Hardware Example 154
8.5.4 Performance Gain 154
8.5.5 Discussion 154
8.6 Case Study: The Parallel Ocean Program 155
8.6.1 Science Goals and Eddy Resolving Simulations 155
8.6.2 Software improvements 156
8.6.3 Performance Gain 156
8.6.4 Discussion 157
8.7 Q3, FY04 157
8.8 Q4, FY04 157

List of Figures

1 Hydroxyl radical in a turbulent jet flame 28
2 Scalar dissipation rate in a turbulent jet flame 29
3 Program flow diagram for S3D 35
4 Performance improvements on Seaborg IBM SP at NERSC . . . 38
5 Performance improvements on Pheonix Cray X1 at ORNL 40
6 Code scaling on IBM SP 42
7 Code scaling on Cray X1 43
8 Mechanical to scalar timescale ratio for mixture fraction versus time. 47
9 Mechanical to scalar timescale ratio for H, H_2 and CO_2 mass fractions and mixture fraction versus time. 48
10 Mechanical to scalar timescale ratio for HO_2, H_2O_2, O and OH mass fractions and mixture fraction versus time. 49
11 a) HO2 mass fraction on a color scale b) OH mass fraction on a color scale. White contours of χ. 50

12 Iso-surfaces of dissipation rates. Blue: χ. Red: χ_{HO2}. Green: χ_{OH}. 51
13 (a) Volume rendering of vorticity magnitude (red, yellow to blue is decreasing intensity) (b) Volume rendering of OH dissipation rate (red, yellow to blue is decreasing intensity) 52
14 Code scaling on various platforms as of Q4-2005. 55
15 Execution time versus number of processors on various platforms as of Q4-2005. 56
16 The elements of rf theory been studied independently, or only loosely coupled. 58
17 Timings for different sized grids as a function of phases -see Table[7]. There are two plots for the different grid sizes measured: one is the actual time to compute the phase and the other this time normalized to the fastest system. In the bottom plot, the y-axis is not labelled but should be time(seconds). That plot is a replica of the 128×128 plot except that a 2000pe run is included where a 200×200 grid is used. 71
18 The fast deuterium distribution function in DIII-D shot 122080 at f = 116 MHz. Contours are shown along the minor radius of the tokamak with the center of the plasma at the lower right side of the image and the edge of the plasma at the upper left. 78
19 A prototype of the TTF 9-cell superconducting accelerating cavity for the ILC linac showing one HOM coupler at the input end (Left) and another HOM coupler together with the fundamental mode coupler at the output end (Right). 88
20 Schematic showing the suite of parallel electromagnetic codes developed at SLAC under the SciDAC Accelerator Simulation project. 89
21 (Left) A distributed ***Omega3P*** model of a quadrant of the DDS, (middle) Microwave QC of the fabricated cells showing measured data within 0.01 percent of target frequency, (right) fabricated DDS cells based on dimensions generated with ***Omega3P***. . . . 90
22 A model of the 55-cell DDS including input/output fundamental mode couplers and HOM couplers for damping wakefields due to HOMs. 90
23 Comparison of DDS wakefields from frequency-domain ***Omega3P*** and time-domain Tau3P simulations. 91
24 An open cavity coupled to three external waveguides with cutoff wavenumber k_{c1}, k_{c2} and k_{c3}. The couplers are denoted by the gray patches. 93
25 Schematic of eigensolvers and linear solvers implemented in ***Omega3P*** 94
26 Shematic for library dependency in ***Omega3P*** 95
27 Workflow diagram in ***OmegaP*** 96
28 Model of the TTF cavity for the ILC 98
29 Electric field distribution of the accelerating mode in the TTF cavity. 99

30 Q_{ext} results between Omega3P and measurements for the two polarizations of the eigenmodes in the first dipole band of the TTF cavity for the ILC. 99
31 Electric field distribution of a HOM in the input (left) and output (right) end of the TTF cavity, showing details of the coupling loops. 99
32 Model of the TTF cavity for the ILC 100
33 Distributed mesh of the ILC LL cavity partitioned with ParMETIS. 100
34 External Q results for 4 monopole bands(left) and 5 dipole bands (right) in the LL cavity design for the IL 101
35 102
36 103
37 CAD model of the SST design for International Linear Collider. . 104
38 Mode patterns of trapped dipole modes in the SST with respective frequency and external quality factor. 104
39 Snapshot of a LAMMPS simulation of a solvated RuBisCO protein and its binding pocket. The two halves of the protein dimer appear in blue and gray; different portions of the binding pocket gate are in yellow and green; the ribulose substate ligand is in orange, a CO_2 molecule is in red; surrounding water has been removed from the image. 111
40 Azimuthal surface velocity profiles for packs of varying height in terms of particle diameter d. Experimental results for glass beads (left) and simulation (right) are shown. 112
41 Parallel efficiency of LAMMPS on 7 machines for the rhodopsin protein benchmark. Fixed-size efficiency (left) and scaled-size (right) are shown. The one-processor timing for the simulation on each machine is shown in parenthesis. 114
42 Parallel efficiency of LAMMPS for the metallic solid benchmark. Same format as the previous plot. 115
43 Parallel efficiency for the granular chute flow benchmark. Same format as the previous plot. 115
44 The speed-up of DOM and DTM on two platforms. All quantities are normalized by the CPU time for DTM with a single processor on Seaborg. 125
45 Inclusive percentage of CPU time consumption of each subroutine averaged over all processors using DOM with 64 processors on Seaborg. 125
46 Inclusive percentage of CPU time consumption of each subroutine averaged over all processors using DTM with 64 processors on Seaborg. 126
47 temperature (flood) and vorticity (black lines) - left to right, t = 0, 3, 5, and 20 msec. 127
48 soot volume fraction, f_ν, with the optically thin radiation model -left to right, t = 0, 3, 5, and 20 msec. 127
49 Temporal variations of the local maximum soot volume fraction, f_ν and soot number density, N_{soot} 127

50 Temporal variations of the volume-integrated soot volume fraction. 128
51 Temporal variations of the integrated soot number density and flame volume. 128
52 Effect of droplet number on the CPU time of spray module with 64 processors 134
53 CPU time usage on Cheetah with 64 processors 135
54 CPU time usage on Seaborg with 64 processors 135
55 Velocity contours at t = 2.46 ms for three grids: 256×128(top),512×256(middle), 1024 × 512(bottom) 136
56 Fuel spray distribution (left) and mixture fraction contour (right) at t=2.46 ms for the case with 1024 × 512 grid points 136
57 Fuel spray distribution at t=0.76 ms for two cases with initial droplet radiuses of 10 mm (left) and 5 mm (right). 137
58 Net wall-clock time used on radiation solver (DTM or DOM). . . 138
59 The speed-up of DTM and DOM with number of processors. . . 139
60 The total radiative power and relative error of OTM, DTM with different number of rays and DOM with different S_n approximations. 139
61 Evolution of the turbulence kinetic energy (TKE) with two treatments 140
62 Turbulence energy spectra at t=0.0055 ms 141
63 Temporal variations of total radiative heat loss in OTM and DOM 143
64 Temporal variations of the integrated soot volume fraction for Cases A-C. 143
65 Temporal variations of the integrated soot number density for Cases A-C. 144
66 Temporal variations of temperature with the optically thin radiation model. From left to right, t = 0, 10, 15, and 20 msec. . . . 144
67 Temporal variations of soot volume fraction, f_{mu}, with the optically thin radiation model. From left to right, t = 0, 10, 15, and 20 msec. 145
68 Temporal variations of soot number density and flame volumes . 145
69 Velocity contours at times 0.56 and 0.72 ms after the start of injection for a 2D spray jet flow with 512 × 256 grid points. The domain size is 2 cm by 1 cm, and the mass flow rate of fuel spray is 0.125 g/s. The spray droplet diameter is $10\mu m$ and the injection velocity is 100 m/s in x direction. Inflow: hard inflow boundary conditions; outflow: nonreflecting condition based on the NSCBC approach [[130]]. 147
70 Velocity contours at times 0.56 and 0.72 ms for the spray jet flow. Initial conditions are the same as in Figure[69].The improved inflow and outflow boundary conditions [[129]] were used. 147
71 Single processor comparison over platforms for the column radiation model. 152

72 CAM 2.0 in the current fiscal year on the IBM p690 system at ORNL. Distributed and shared memory optimizations are studied here. 155
73 The evolution of performance of the POP code due to software and hardware improvements on the Cray X1 over the last year is shown. 156

List of Tables

1 Sample S3D INCITE result for FY2005: The metric of interest in this enhancement is (compute time / gridpoint) / time step . . 15
2 Sample AORSA results for FY2005: references into the text are in parentheses 16
3 Sample CCSM result for FY2005: the metric of interest in this enahncement is labeled *stepon* in table[8] 16
4 Sample Omega3P result for FY2005: The metric of interest in this enahncement is (compute time / eigenmode) / processor. . . 17
5 AORSA Start-Up Data File Sizes 67
6 AORSA Output Sizes 67
7 Major phases of aorsa2d for referencing Figure[17] 72
8 Cray X1E with 128 MSPs, collected September 2005 85
9 CCSM3 T85 on Cray X1 with 248 MSPs 87
10 Execution time and memory used for a moderate-size quadratic eigenvalue problem solved with SOAR/MUMPS on 32 MSPs of the Cray X1 at ORNL. 102
11 On Pheonix 32 MSPs were used to calculate 18 eigenpairs. On Seaborg 32 nodes were used to compute the 18 eigenpairs. There were 2 tasks per node and 8 threads per task on the IBM runs. . 105
12 CPU time for 100 timesteps of the rhodopsin protein benchmark on the Liberty machine with short-range Coulombic tabling and rRESPA hierarchical timestepping options turned off and on. . . 117
13 CPU time used by the spray module as a percentage to the total CPU time of S3D 134
14 Performance of various radiation models in terms of CPU time/time-step/grid/processor. Units are seconds. 148

Improve Computational Science Capabilities. *Average annual percentage increase in the computational effectiveness (either by simulating the same problem in less time or simulating a larger problem in the same time) of a subset of application codes within the Scientific Discovery through Advanced Computing (SciDAC) and base program efforts.* **FY05 target: 50%**

1 Overview

The goal is to measure the improvement of the computational science capabilities for a subset of Scientific Discovery through Advanced Computing (SciDAC) and base applications from the Department of Energy's Office of Science (DOE SC).

In the first quarter of FY2005 a long list of compute bound or driven applications was reduced to a short list by committee. The applications selected for this year's report represent a broad base of science endeavors: combustion, fusion, climate, accelerator physics, and (classical) molecular dynamics. It is noteworthy that none of the application groups has received explicit funding to provide the information presented in this report.

To understand the evolution of the capabilities of the application software efforts over the course of the fiscal year, a scientific baseline was formed in the second quarter for each individual application. The baseline information includes the measured time (data) required to compute answers to current physical questions or to simulate relevant physical systems on one of several SC hardware platforms. Ideally, the baseline also includes information (detailed) describing *why* the physical questions posed in the case studies are important to ask and answer and *how* this is achieved computationally. The baseline is the reference material for the subsequent enhancements made to the individual efforts during the fiscal year.

The effectiveness will be determined by quantifying science or performance enhancements made to the baseline software versions. The enhanced applications may resolve the current science *more physically* in the same or similar compute time as the reference application, may compute results to new physical problems, or may compute results faster than the reference application on a common problem. Both the enhanced application and the baseline application are reported on the same hardware platform to possibly filter out the impact hardware dynamics has on the enhancements. Also, strong scaling on the system is ruled out as an enhancement. It is noted that by constraining the study to a *fixed* hardware platform, many of the performance enhancements made may not translate to different platforms. Further, due to forces not within the control of this inquiry, some of the hardware platforms used to generate the baseline have changed this fiscal year making idealized comparisons impossible.

1.1 Determining software effictiveness

What is the constraint? The target for FY2005 is to assert a 50% increase in the computational effectiveness of the reported applications.

Did the applications in this report satisfy the constraint, **yes** or **no**? To answer this question requires an interpretation of the target and the information generated by the application groups. The interpretation follows.

There will be two modes of satisfying the constraint(s) (**yes** answers) for this year's goal. A *strong* pass denotes a case that is empirically quantifiable. It is most likely affiliated with a performance enhanced application. A *weak* pass denotes a case that is not easily quantified. It is most likely affiliated with science enhancements to the application and may require a third party from the application field to assess the relevance of the enhancements.

In this report, if either of the following two cases are satisfied, it is a pass for the application.

1. $\frac{(T_o - T_1)}{T_o} \geq .50$ (strong pass):

   - T_o is the measured time required for $code_k$ to compute $problem_i$ on $system_j$; $code_k$ is the original application software required that accepts input and generates output for the physical problem instance; $problem_i$ is the physical problem instance; $system_j$ is the subset of the hardware platform used for computing results -e.g. the hardware instance
   - T_1 is the measured time required for $code_{k'}$ to compute $problem_i$ on $system_j$; $code_{k'}$ is the performance enhanced version of $code_k$ and thus computes $problem_i$ on $system_j$ faster than $code_k$

   There may exist variations in the performance due to different hardware instances. As an example, if the system has 10 processing elements then there are n = 120 unique allocations with 3 processing elements. The times recorded in this report reflect those measured on a specific instance of the hardware platform -e.g. one of the 120 possibilities. It is not trivial to exactly replicate a hardware instance on many of the hardware platforms since there is a scheduler that determines hardware selection based upon hardware requests and availability. Further, it may not be necessary if the system behaves normally when computing the problem instance. The normal behavior would be corroborated when the statistical variation in the compute time for the problem instance with the 120 possible hardware allocations was *small* when compared to the mean compute time over the same hardware instances. A normally behaving system has been assumed.

2. *epic* or $T_1 \sim T_o$ (weak pass):

   - T_1 is the measured time required for $code_{k'}$ to compute $problem_{i'}$ on $system_j$; $problem_{i'}$ is a new physical problem instance (*epic*) *or* a bigger, more physical version of $problem_i$; $code_{k'}$ is the science

enhanced version of $code_k$ capable of computing $problem_{i'}$; $code_k$ cannot compute $problem_{i'}$ (thus confusing the interpretation of this metric) but can compute a similar yet *smaller* problem on the same $system_j$

Again, there seems to be no clear way to assign a value to *new science capabilities*. In such cases, judgements were defered to the respective application communities.

1.2 Results

The reader is instructed to review the table of contents for each of the applications. It is noteworthy that several of the applications made enhancements on more than a single hardware platform. This does not contradict the fixed hardware platform constraint -rather indicates a concurrent effort over platforms.

Rather than focus on what the report is not, the high level results now follow. The corroborating details and the implications of the enhancements are described in the individual sections of the support text. Pointers into the text are provided.

1.2.1 S3D INCITE

- **Code name:** S3D: The code solves the compressible Navier-Stokes, total energy, and species continuity conservation equations in multi-dimensions using high-order spatial finite difference discretization and high-order Runge-Kutta explicit/implicit temporal integration on a uniform or stretched mesh.
- **Scientific field - relation to SC Program:** The code is primarily used to model (turbulent) combustion chemistry. The SC program that the research is closest to in scope is BES.
- **Primary Technical Contact:** Jacqueline Chen, Sandia National Laboratories

Problem	System	Enhancement
CO-H_2 chemistry benchmark (sec2.10)	Cray X1, *phoenix.ccs.ornl.gov*	**performance**, $\sim$ 57.74% (sec2.11.2,sec2.14.2)

Table 1: Sample S3D INCITE result for FY2005: The metric of interest in this enhancement is (compute time / gridpoint) / time step .

The results in the support section for S3D INCITE suggest a strong pass on the Cray X1E architecture. This is likely true but was conjectured based upon times measured on both the Cray X1 and X1E. Because the times for the theoretically comparable runs were not measured on the same hardware system -the substantial gains reported are not presented in this summary. However, the group certainly can claim a strong pass based upon the baseline numbers

generated at the end of Q2 (3.55e-8 hours/gridpoint/timestep) versus the time reported at the end of Q3 (1.5e-8 hours/gridpoint/timestep).

1.2.2 AORSA

- **Code name:** AORSA, *A*ll-*OR*ders *S*pectral *A*lgorithms, solves the wave equation for radio frequency heating in multi-dimensional plasmas with no restriction on wave length relative to orbit size and no limit on the number of cyclotronic harmonics.
- **Scientific field - relation to SC Program:** The SC program that the research is closest to in scope is FES.
- **Primary Technical Contact:** Don Batchelor , Fred Jaeger , Oak Ridge National Laboratory

Problem	System	Enhancement
NSTX benchmark (sec3.4)	IBM SP Power3, *seaborg.nersc.gov*	**performance**, $\sim$ 53.05% (sec3.7.2)
	Cray X1, *phoenix.ccs.ornl.gov*	**performance**, $\sim$ 55.75% (sec3.7.2)

Table 2: Sample AORSA results for FY2005: references into the text are in parentheses

It is worth noting that AORSA, like some other applications, reports the effectiveness in terms of a speedup factor, call it γ. This means that the time to compute the results with the original code is γ times as long as the code that has been enhanced for performance. That is $T_o = \gamma T_1$ and thus the assertion becomes $\frac{T_o - T_1}{T_o} = \frac{\gamma T_1 - T_1}{\gamma T_1} = \frac{\gamma - 1}{\gamma}$. Clearly, when $\gamma \geq 2$ then the constraint is satisfied.

1.2.3 CCSM

- **Code name:** Community Climate System Model, CCSM,
- **Scientific field - relation to SC Program:** The code is primarily used to model the earth's climate dynamics with the coupling of the atmosphere, sea ice, land, and ocean components. The SC program that the research is closest to in scope is BER.
- **Primary Technical Contact:** John Drake, Oak Ridge National Laboratory ; Phil Jones, Los Alamos National Laboratory

Problem	System	Enhancement
CAM3.0 w/ FV C-grid (sec4.4)	Cray X1E, *phoenix.ccs.ornl.gov*	**performance**, $\sim$ 53.7% (table[8])

Table 3: Sample CCSM result for FY2005: the metric of interest in this enahncement is labeled *stepon* in table[8]

1.2.4 Omega3P

- **Code name:** Omega3P: based mainly on the finite-element formulation, the code suite solves Maxwell's equations in the time and frequency domain with or without particles.
- **Scientific field - relation to SC Program:** The code is used to study and design various cavities for the International Linear Collider -a particle accelerator facility. The SC program that the research is closest to in scope is HEP.
- **Primary Technical Contact:** Kwok Ko, Stanford Linear Accelerator Facility

Problem	System	Enhancement
model HOMS in ILC cavities (sec5.5,sec5.6)	IBM SP Power3, *seaborg.nersc.gov*	**performance**, $\sim$ 81.3% (sec5.5.3,table[11])

Table 4: Sample Omega3P result for FY2005: The metric of interest in this enahncement is (compute time / eigenmode) / processor.

Since the analysis is not provided in the Omega3P section, it will be provided here.

The science driven measure of performance for the Omega3P code is (compute time / eigenmode) / processor. The first observation is that the metric is generic across the different physical problems studied this FY with Omega3P.

The original code employs 743K tetrahedral mesh elements and linear basis functions with the self-consistent loop to resolve HOMS in the ILC TTF cavity. The metric for this run is 1.23 seconds / eigenmode / processor as measured on Seaborg at NERSC.

The enhanced code at the end of the FY computes the metric to be (see table[11]) .229 seconds / eigenmode / processor. It is noteworthy that the problem modeled at the end of the fiscal year employs 523K tetrahedral mesh elements with quadratic basis functions and second-order geometry with the Second-Order Arnoldi routine to resolve HOMS in the ILC LL cavity. Thus, not only was the code enhanced for performance but also the science capability was increased by a factor of $\sim$ 3.89 . This factor is related to the number of degrees of freedom that can be resolved with the original and enhanced codes. In the original code, 743K tetrahedral mesh elements and the linear basis functions equates to 816337DOFs while the new code permits 3181290DOFs.

1.2.5 LAMMPS

- **Code name:** *L*arge-scale *A*tomic/*M*olecular *M*assively *P*arallel *S*imulator, LAMMPS, is a classical molecular dynamics simulation code.
- **Scientific field - relation to SC Program:** The SC programs BER and BES represent the scope of research studied with the code.

- **Primary Technical Contact:** Steven Plimpton, Sandia National Laboratories

A **weak** pass will be argued for LAMMPS here.

Residual stress in thin film structures is known to dramatically influence device performance. To study this phenomena it is necessary to perform simulations of islands on substrates where the island is metal and the substrate is either metal or oxide. (see sec6.4.2) Before this FY, LAMMPS could only compute the former case.

The **weak pass assertion** here is that the tasks for hybrid pair and bond potentials, pre-tabulated pairwise potentials, and class two inter-molecular potentials were implemented this FY along with various other pair potentials (Yukawa, Morse, Buckingham). The enhanced application permits arbitrary control over the strength of interaction between island and substrate (i.e. island adhesion). The capability will allow isolated islands of varying sizes to be simulated to extract the dependence of stress in the island on island size and adhesion to the substrate. When islands grow large enough to impinge upon one another, island coalescence is theorized to be a primary contributor to stress generation.

1.2.6 S3D SciDAC

- **Code name:** S3D: The code solves the compressible Navier-Stokes, total energy, and species continuity conservation equations in multi-dimensions using high-order spatial finite difference discretization and high-order Runge-Kutta explicit/implicit temporal integration on a uniform or stretched mesh.

- **Scientific field - relation to SC Program:** The code is primarily used to model (turbulent) combustion chemistry. The SC program that the research is closest to in scope is BES.

- **Primary Technical Contact:** Hong Im, University of Michigan

The improvements for the current FY have enabled both new science and performance enhancements that satisfy for both a **weak** and a **strong** pass. Since it is relevant to argue the new science case first, the assertions are outlined here with references into the S3D SciDAC section.

The physical problem computed is a spray jet simulation with the same initial flow conditions and domain size as described in Figure[69].

The original version of the application employs the conventional hard inflow boundary conditions and the Navier-Stokes characteristic boundary conditions (NSCBC) [130] for treating the outflow boundaries. For combustion cases, Sutherland and Kennedy [131] further improved the treatment by considering the chemical reaction source terms at the domain boundaries. Despite this improvement, the previous boundary condition treatment have revealed a significant level of spurious acoustic waves at the inflow and outflow boundaries

as the spray jet reaches the outflow boundary. The solution fields computed with the original BCs are shown in Figure[69].

The enhanced version of the application reported here employs new characteristic boundary conditions [129] to mitigate spurious (numerical) acoustic waves at the boundaries. The solution fields computed with the physically improved BCs are shown in Figure[70].

The **weak pass assertion** here is that the spurious acoustic waves present in the original BCs are completely eliminated with the improved BCs. The level of improvement in the solution fidelity is unprecedented.

The **strong pass assertion** here is 75% on the MPP2 HP Integrity system at PNL (*mpp2e.emsl.pnl.gov*). The enhancement comes from the fact that the improved boundary conditions for the jet spray simulations allow a smaller number of grid points to accurately resolve the spurious acoustic waves. The original code requires at least a factor of 2 more grid points in each dimension to achieve similar physical accuracy. The added cost for the original code is manifest in both added memory and cpu computing expenses.

1.3 Support Information

The the rest of this report is dedicated to describing the baseline problems and subsequent evolution of the capabilities of the applications cited in the results. There were far too many weak and strong pass cases for the applications in this year's effort to enumerate them in this summary. This summary aims merely to justify the assertion that each application satisfied the software improvement constraint defined at the beginning of the document.

2 S3D : Innovative and Novel Computational Impact on Theory and Experiment (INCITE)

Executive Summary

In anticipation of performing the FY05 INCITE goal, the first Direct Numerical Simulation (DNS) of a 3D turbulent nonpremixed flame with detailed chemistry aimed at understanding extinction and reignition mechansisms, we have successfully optimized key kernels in the DNS code, S3D, on NERSC's IBM SP, Seaborg, and on ORNL's Cray-X1, Phoenix. This document provides an overview of the INCITE goal, the mathematical formulation and numerical implementation, the rationale behind the selection of the physical configuration and parameters, and code optimization on two platforms. While optimization on Seaborg resulted in modest gains, vectorization of S3D on the Phoenix resulted in ten-fold increases in performance efficiency. S3D was found to scale well on both platforms.

2.1 Introduction

The objective of this document is to provide an overview of the code optimization and performance studies leading up to and including the FY05 INCITE goal as part of the 2005 Joule Software Effectiveness Study. The INCITE goal is to simulate directly the first 3D turbulent nonpremixed $H_2/CO/N_2$-air flame with detailed chemistry aimed at studying the mechanisms of extinction and reignition. The grid number planned is 0.2 billion. To our knowledge, this would be the largest DNS of a turbulent flame with detailed chemistry performed to date. A state-of-the-art parallel 3D Direct Numerical Simulation (DNS) solver for turbulent reacting flows, S3D, is used to perform the simulations [67]. This code was developed with support from the BES Chemical Sciences over the past decade, and more recently, has been extended to include multi-physics and numerical improvements by the SciDAC project High-Fidelity Terascale Simulations of Turbulent Combustion. S3D is a F90/F77 code that is extensible and scales well to over 4000 IBM SP processors using MPI for scaleable parallelism. The code solves the compressible Navier-Stokes, total energy, and species continuity conservation equations in multi-dimensions using high-order spatial finite difference discretization and high-order Runge-Kutta explicit/implicit temporal integration on a uniform or stretched mesh. This document provides the background and motivation for the INCITE goal, the specific research objectives, a description of the mathematical formulation of S3D, the computational approach, physical configuration, simulation parameters, and S3D code improvements on NERSC's IBM SP and ORNL's CrayX1.

2.2 Background and Motivation

In many practical combustors the fuel and air are not premixed. For example, this is the case in aircraft applications where fuel and oxidizer are segregated for safety reasons, and in direct injection internal combustion engines, for reasons related to efficiency gains. Therefore, many fundamental combustion questions revolve around the issue of rapid mixing of the reactants which is desirable to maximize heat release rates, thus enabling smaller combustion chamber volumes, and minimizing the production of pollutants. The disadvantage of enhanced mixing rates is that, above a critical value, local extinction or even destabilization of the entire flame can occur. Extinction is dependent mainly upon the balance between local mixing and chemical rates, which depend, in turn, upon the local fuel-air composition and temperature. Extinction adversely affects efficiency, pollutant generation and safety. If extinguished pockets of unburned mixture fail to reignite during a given combustion residence time, then reactions are quenched and unburnt fuel is emitted out of the combustor. If extinguished pockets are abundant due to excessive turbulent strain, and reignition is slow, combustion may cease altogether. In an aircraft, of course, this would be catastrophic.

DNS of turbulent reacting flows has long been a useful, yet computationally limited tool to gain fundamental insight into the physics of turbulence-chemistry

interactions [66,74,95,102]. These interactions reflect the coupling between fluid dynamics, chemistry, and molecular transport in reacting flows. Even within the continuum assumption, where the Navier-Stokes equations are valid for a large class of flows, the range of length and time scales (over 10 decades) may impose prohibitive requirements on high-fidelity, fine-grained simulations, such as DNS, which are beyond present computer capabilities. Therefore, it is generally accepted that the primary simulation tools for design and optimization of combustion devices will remain limited to two coarse-grained approaches, Reynolds-Averaged Navier Stokes (RANS) simulations and Large-Eddy Simulations (LES). In RANS, the Navier-Stokes (N-S) equations are solved for ensemble mean quantities; while LES is based on the spatial filtering of these equations such that a range of small length and time scales, notably those scales where strong turbulence-chemistry coupling occurs, is not resolved in the simulations. The unclosed terms that result from averaging or filtering of non-linear terms in the incompressible N-S equations require additional modeling. In reacting flows, additional modeling is required for transport and chemical source contributions in the species and energy equations that reflect highly non-linear phenomena. Therefore, the correct representation via modeling of the small scale mixing and reaction interactions is crucial to the successful prediction of efficiency, stability, and emissions in practical devices. Principal approaches to non-premixed combustion modeling include those based on the mixture fraction, e.g. steady [92] or unsteady flamelets [94], and conditional moment closure (CMC) [80], those based on the solution of the transport equation for the joint probability density function (PDF) [97], and an approach based on a statistical one-dimensional description of turbulence [79].

As a result of its technological importance, extinction and reignition, and other finite-rate phenomenon in nonpremixed combustion has received considerable attention recently, due in part to a well-documented series of experiments on turbulent jet flames that exhibit local unsteadiness and extinction. The resulting library of flame data has been an invaluable benchmark for the advancement of fundamental understanding and model validation of nonpremixed turbulent combustion in an international collaboration among experimental and computational researchers referred to as the Turbulent Nonpremixed Workshops `http://www.ca.sandia.gov/TNF/abstract.html`. Several groups in this forum have demonstrated reasonable success in modeling nonpremixed flames without extinction. However, there are still limitations and uncertainties in these models in their ability to describe extinction and reignition, and other important finite-rate combustion phenomena. For example, in the ODT model, only one mode of reignition is possible due to its intrinsic one-dimensional nature. Multi-dimensional ignition modes would need to be included in this model empirically - the same comment applies to flamelet approaches. A key limitation in the transported PDF approach is the choice of relevant mixing time scale(s) in the presence of finite-rate chemistry, and the influence of preferential diffusion among species on this selection. CMC approaches may require additional conditioning variables in order to predict extinction and reignition. In summary, current modeling approaches would benefit greatly from more detailed charac-

terization of the dynamics of extinction/reignition and other finite-rate effects in turbulent nonpremixed combustion.

In recent years, the rapid growth of computational capabilities has presented both opportunities and challenges for high-fidelity simulations of turbulent combustion flows. Realistic simulations that address complex multi-physics interactions, such as the so-called turbulence-chemistry interactions in combustion flows, have become accessible through the growth of processor speed, computer memory and storage, and significant improvements in computational algorithms and chemical models. While the opportunity exists for gleaning fundamental physical insight into fine-grained chemistry-flow interactions in simplified two-dimensional physical configurations (see the review in [74]), it remains a formidable challenge to directly simulate three-dimensional turbulent flames with detailed chemistry. In the past several years, the advent of terascale computers in the U.S. and in Japan, has made it possible to begin to study fundamental issues such as flame stabilization and extinction in three-dimensional laboratory configurations with multi-step chemistry using the DNS approach [88, 89, 91]. These simulations are costly, requiring several million cpu-hours on a terascale computer and between 20 and 100 million grid points. While costly, three-dimensional turbulent direct numerical simulations with detailed chemistry enable both turbulence dynamics and chemical reaction to be accurately represented concurrently, thus opening new realms of possibility for the understanding of turbulence-chemistry interactions and the development of models.

2.3 Research Goals

The primary objective of the proposed study is to perform a three-dimensional turbulent direct numerical simulation of a nonpremixed $H_2/CO/N_2$- air flame with detailed chemistry. This simulation, the first in a series of different Reynolds numbers, will be targeted at providing fundamental insight into key outstanding issues related to modeling of turbulent nonpremixed combustion: extinction and reignition, flow and flame unsteadiness, the correlation of strain rate and scalar dissipation rate, differential diffusion of species, and turbulent mixing in a finite-rate chemical environment. Through collaboration with experimentalists and modelers in the TNF Workshop, we also plan to gather statistics required to further improve or validate different modeling approaches. In the following subsections the specific objectives of the proposed work are presented.

2.3.1 Extinction and Reignition Dynamics and Statistics

In the absence of significant preferential diffusion effects, models for non-premixed combustion are presently capable of representing with reasonable accuracy turbulent flows without extinction. However the inclusion of extinction and reignition remains a challenge. There is a need to provide fundamental information regarding the mechanisms of extinction and re-ignition in a turbulent environment. These processes are likely quite dependent on finite rate, complex chem-

istry interactions with turbulence and turbulent mixing. The proposed DNS will make a new contribution to this understanding by including detailed chemistry (i.e. capable of representing fully burning and igniting chemical states), heat release, and realistic thermo-chemical properties. Previous studies have typically either used reduced chemistry with heat release [91], reduced chemistry without heat release and artificial adjustment of rate constants [83] or global one-step chemistry without heat release [86, 87, 99]. Unlike previous DNS studies, the proposed physical configuration with detailed chemistry will permit reignition to occur by either autoignition via chemical chain-branching or by flame propagation (either normal or tangential to the stoichiometric surface of the extinguished flame.) Statistics regarding the occurrence of the different modes of reignition as a function of key flow and flame parameters will be obtained.

2.3.2 Differential Diffusion

Several nonpremixed combustion models parameterize the thermochemical state with a conserved scalar known as the mixture fraction. The mixture fraction is conserved if the species are assumed to have equal mass diffusivity. Hydrogen and hydrocarbons exhibit a wide spectrum of mass diffusivities distinct from thermal diffusivity. Fast-diffusing, chemically crucial intermediates like H atom are mobile and can segregate from other species and heat. Recently, [100] have proposed a method of quantifying the degree of differential diffusion (DD) in a flame. The proposed DNS will be used to begin to understand the significance of differential diffusion, as a function of local mixing and reactive conditions, on finite-rate phenomena such as extinction and reignition. For example, we will seek a conserved scalar definition that is least sensitive to DD and a relevant combustion progress variable definition, that together, may allow us to parameterize extinction and reignition processes in a real flame. In the context of LES, the DNS data will be spatially-filtered to assess the relative importance of DD to convection and sub-filter terms on the filtered mixture fraction equation.

2.3.3 A-priori Model Development and Validation

3D simulations with complex chemistry, and turbulence parameters within the realm of moderate Reynolds number flames will be of significant interest to the combustion modeling community. After an initial investigation of the data, our plan is to invite members of the modeling community to employ the data set either through collaboration or by sharing our data. Several of the key modeling issues that can be addressed with the proposed DNS are outlined below.

For approaches based on the mixture fraction or other conditioning variables, it will be possible to assess the magnitude of the conditional fluctuations, leading to understanding of the extent to which the thermo-chemical state can be parameterized by a reduced set of variables, and, building on our previous work, developing alternative choices for conditioning variables [100]. Furthermore it will be possible to assess the degree to which these variables can be predicted

from the resolved or mean flow, for example using presumed forms of the PDF.

Flamelet approaches may be evaluated and improved by a better understanding of unsteady, multidimensional and differential diffusion effects, and of course extinction and reignition. For flamelet approaches it will be possible by Lagrangian tracking of fluid or flame elements to identify and understand effects of unsteadiness, including extinction and re-ignition, providing valuable information for recently developed approaches to account for these effects [87, 93]. Multi-dimensional effects can similarly be identified using our parallel surface-based post-processing algorithms [73]. Statistics of the conditional average of the scalar dissipation or its PDF can be obtained from the DNS.

CMC approaches will benefit from a better understanding of the magnitude of the conditional fluctuations and the conditional scalar dissipation [80], particularly the differences between extinguished and fully burning regions. In the case of extinction, CMC may require a second conditioning variable, and the DNS can be used to provide information on the selection of the second conditioning variable and the doubly-conditional scalar dissipation rate [83]. Differential diffusion represents a challenge for CMC [81, 82], and DNS can contribute to its development. Recently the CMC approach has been applied to LES, where it is noted that a full implementation of the CMC on the LES grid may be prohibitive due to the introduction of the additional mixture fraction dimension. However, researchers [64, 90] have argued that the spatial variations of the conditional averages may be significantly less than for unconditional quantities, potentially allowing the use of a coarser grid for LES simulations. It will be possible to verify the validity of these assumptions, and any dependence on filter size.

For transported PDF approaches, the main closure problem is for scalar mixing [97]. Presently PDF approaches do not distinguish between mixing of conserved and reacting scalars. However reacting scalars can potentially have very different mixing characteristics, particularly where there are different diffusivities and extinction [86]. Also there may be different mixing characteristics of the conserved scalar in extinguished and burning regions. It will be possible to make a contribution to the understanding of these issues through study of the DNS database.

The disadvantage of the ODT model is that it allows only limited mechanisms for reignition due to its intrinsic one-dimensional nature. For example, triple flame or edge flame propagation which requires nonaligned gradients of the mixture fraction and progress variable can not occur along a one-dimensional domain and would need to be empirically modeled [76]. The DNS could assess the importance of the different reignition modes, and provide clues towards empirical models for edge and triple flame propagation.

2.3.4 DNS Benchmark for comparison

This use of the data is inspired by the highly successful TNF Workshop, in which experimental benchmark flames were developed and significant progress was made in model development through the provision of a collaborative framework for comparison of modeled and measured results. Typically DNS is not used in

this way, rather it is normally performed and exploited by a single or limited number of research groups seeking to advance a particular modeling strategy. However, there are good reasons to suggest that this could be a very profitable use of DNS data. While DNS is necessarily performed with only a limited range of length scales, the ambiguities present in an experiment, in terms of comparisons of modeled and measured results, simply do not exist. The thermochemistry, boundary and initial conditions are all completely known, and there is negligible measurement error. Furthermore in the DNS we will have access to time dependent three-dimensional fields, which greatly enhances possibilities for evaluation of models beyond the typical comparisons of simple point-wise averages. It is proposed that after an initial investigation of the results, the data set will be introduced to and shared with the community under the framework of the TNF workshop. Our target for this would be the forthcoming TNF workshop in 2006.

2.3.5 LES Connection

The proposed work is planned in conjunction with a simultaneous LES effort at Sandia led by J. Oefelein. In contrast to DNS, LES allows a high fidelity representation of the large scales that are strongly influenced by the geometry of the problem, and small scales are modeled. The LES and DNS efforts are complementary and allow us to span the range of scales that exist in a laboratory flame. It is planned to exploit the LES to provide more realistic boundary and initial conditions for DNS, and for the DNS to provide improved sub-grid scale models for the LES.

2.3.6 Experimental Connection

We plan to use DNS data to determine the effect of measurement uncertainties, for example, due to photon shot noise, on measurements of the scalar dissipation rate. Comparison between experiment and DNS will be achieved through spatial filtering and by modeling shot noise in the DNS data and comparing with the raw DNS and experimental data. Similar comparisons have already been made with large-eddy simulation of a jet flame; however, in the LES approach the full spectrum of mixing and combustion is not resolved on the grid, but rather modeled through flamelets [71]. Disparities in such a comparison may not entirely be attributable to shot noise. We will further examine the adequacy of the OH radical to extract flame normal vectors required for multi-dimensional scalar dissipation rate measurements [77].

2.4 Computational Approach

The simulation will be performed using Sandia's massively parallel direct numerical simulation code, S3D. This code solves the full compressible reacting Navier-Stokes, total energy, species and mass continuity equations coupled with detailed chemistry. It is based on a high-order accurate, non-dissipative

numerical scheme. It has been used extensively to investigate fundamental turbulence-chemistry interactions in combustion topics ranging from premixed flames [66, 73], autoignition [69, 98], to nonpremixed flames [85, 100]. Time advancement is achieved through a six-stage, fourth-order explicit Runge-Kutta (R-K) method [78], spatial differencing is achieved through high-order (eighth-order with tenth-order filters) finite differences on a Cartesian, structured grid [78], and Navier-Stokes Characteristic Boundary Conditions (NSCBC) [96, 101] were used to prescribe the boundary conditions. The equations are solved on a conventional structured mesh, and scaleable parallelism is achieved through MPI.

This computational approach is very appropriate for the problem selected. The coupling of high-order finite difference methods with explicit R-K time integration make very effective use of the available resources, obtaining spectral-like spatial resolution without excessive communication overheads and allowing scalable parallelism. An alternative strategy that could have been employed is the use of Adaptive Mesh Refinement (AMR). While AMR is attractive for many combustion problems, it is very doubtful that this approach would result in computational savings in the present case. AMR is most efficiently applied in cases where there is a large disparity between flame and turbulence length scales. In the present case, however, the flame and turbulence length scales are overlapping, and thus the region in which a fine grid is required occupies a large proportion of the computational domain. In addition, AMR has not yet been demonstrated to scale up to the large number of processors required for a calculation of this magnitude.

2.5 Physical Configuration

DNS of a 3D turbulent nonpremixed $CO/H_2/N_2$-air jet flame with detailed chemistry will be performed. The kinetic mechanism employed for CO/H_2 oxidation includes 12 species and 33 reactions [84]. The physical configuration chosen corresponds to a temporally-evolving plane jet flame. In the temporal configuration an inner turbulent fuel core flows within quiescent air, and these streams are separated by reacting mixing layers under the influence of significant mean shear. The configuration results in similar but not identical turbulent structures to those observed in a spatially evolving planar jet, with an observation window that moves with the mean jet velocity. This configuration was selected rather than the spatially-evolving jet because it allows for more significant flame-turbulence interaction within a given computational domain with wider separation in mixing scales than previously possible [91, 99], thereby potentially creating a more wrinkled flame surface through intense turbulent mixing. More wrinkling may lead to a greater probability of different modes of reignition to occur than previously studied, such as by flame propagation in a direction normal to the stoichiometric surface or by self-ignition of a fluid parcel following heat conduction from neighboring flame or product gases. Pantano focused primarily on reignition via edge flame propagation tangent to the stoichiometric surface [91]. The other modes rely on high strain to reduce the

separation distance between burning and quenched regions in a wrinkled flame, in the direction normal to the stoichiometric surface, so that heat conduction and radical diffusion can reignite the mixture.

To ramp up to the large INCITE calculations, many test calculations ranging from three to six million grid points and one production calculation with forty million grid points have been performed. The production calculation was performed on the MPP2 Linux cluster at PNNL and run on 480 processors there. Figure 1 shows a volume rendering of the hydroxyl radical mass fraction at an instant at approximately 4 flow-through times into the simulation. The field shows a complex three-dimensional structure with areas of low OH. Figure 2 shows a volume rendering of the local scalar dissipation rate. Areas of high scalar dissipation, highlighted in red, exist in highly localized sheet-like structures, which also correspond to low OH values. This calculation reveals significant three-dimensional flame structure largely generated by vortex stretching induced by the mean shear, and localized regions of extinction followed by reignition.

Relative to this run, we expect to increase both Reynolds numbers and the amount of extinction for the INCITE calculation.

2.6 Physical and Numerical Parameter Selection Considerations

The Direct Numerical Simulation of a reacting jet with extinction and re-ignition is a formidable task, both in terms of the computational cost and in finding the optimal physical parameter space. The numerical and physical parameter selection is closely intertwined and must be discussed together.

The computational cost of this explicit code can be easily estimated as the product of the total grid number, the total number of time steps and the cost per grid point and time step. It is essential to resolve both the small scale chemical and fluid mechanical timescales as well as provide enough large scale structures to allow for normal flow development and adequate statistics. There are limitations on the total number of grid points and on the total computational cost.

Selection of the physical parameters is determined mainly by the Reynolds number, which it is desirable to maximize, and the characteristics of extinction and reignition. First, an appropriate amount of extinction must be obtained. This is governed largely by the Damköhler number, defined as the ratio of a characteristic turbulence timescale to a characteristic flame timescale. Low Damköhler numbers are required to obtain extinction. Second, realistic reignition modes must be obtained. This implies that mixing rates must relax in order to allow reignition, which occurs naturally in the plane jet configuration, but must occur within a reasonable computational time. We believe that the importance of different reignition modes is governed by the ratio of the turbulence intensity to the edge flame speed, introducing a further parameter of importance.

The number of grid points is determined by simultaneous considerations of the Reynolds number, Damköhler number, and resolution required by the chem-

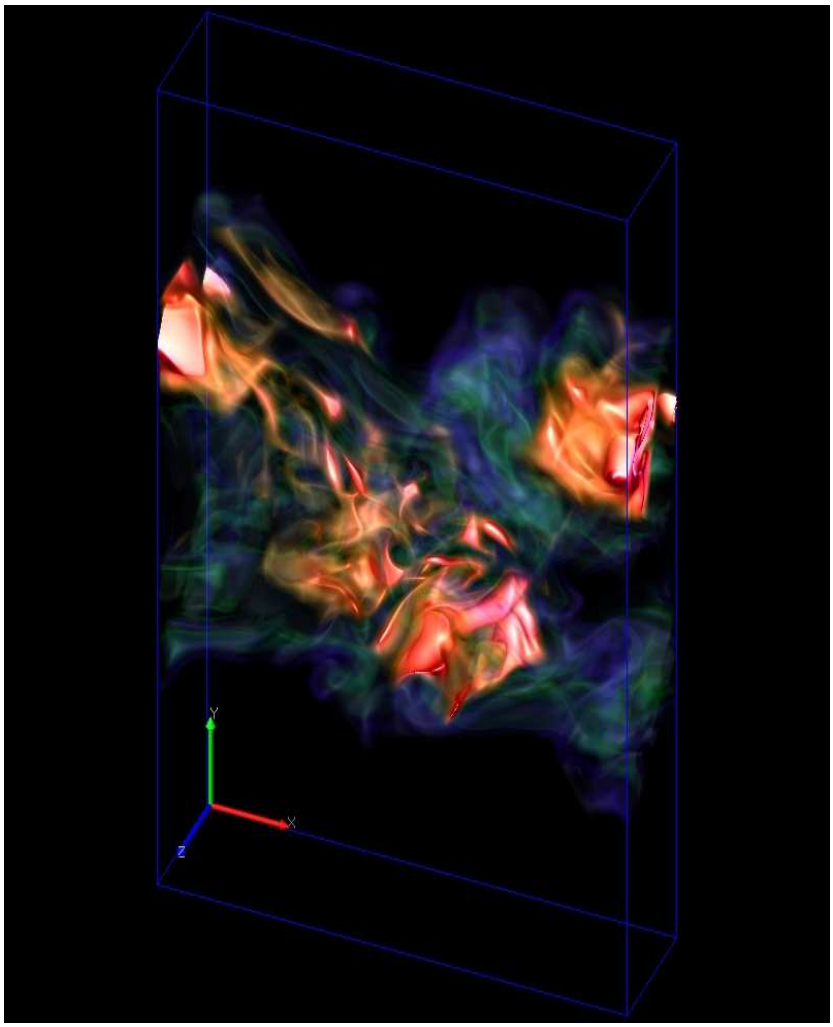

Figure 1: Hydroxyl radical in a turbulent jet flame

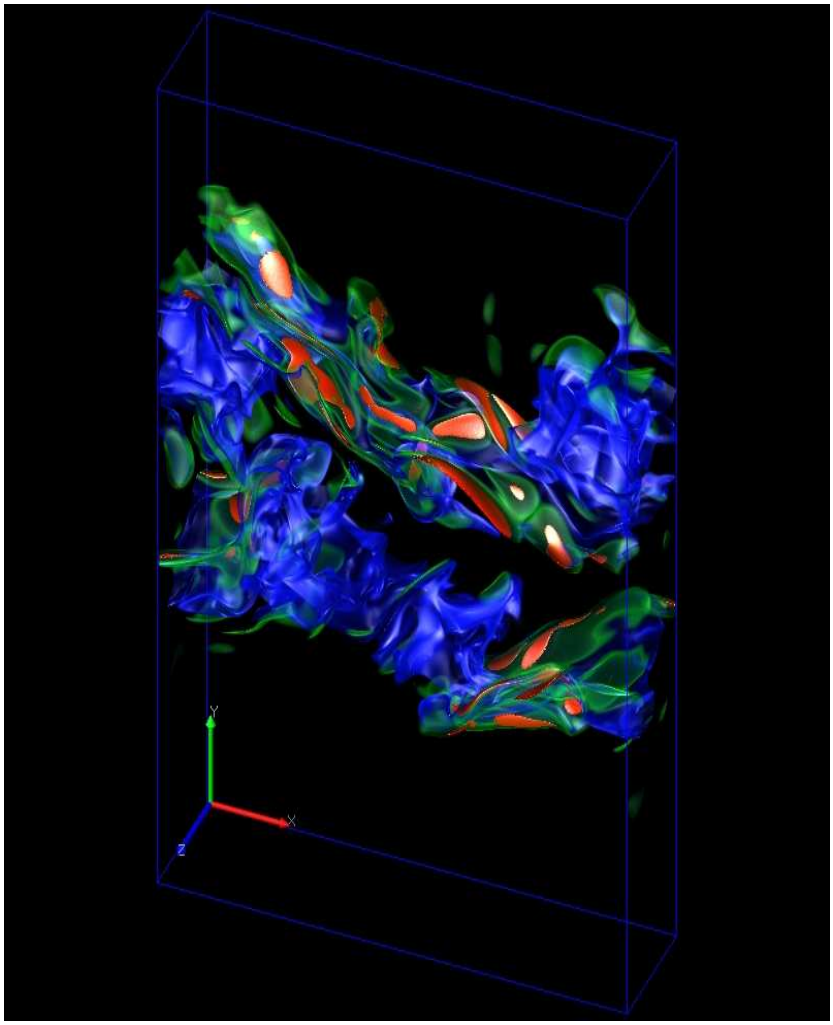

Figure 2: Scalar dissipation rate in a turbulent jet flame

istry. It is well known that the resolution requirements for cold flow DNS scale with $Re^{9/4}$. The Reynolds number together with an adequate number of large scale structures implies a certain grid resolution, and the Damköhler number together with an adequate range of mixture fraction scales also implies a resolution. Reynolds number scales with the jet velocity and height, while Damköhler number is proportional to the height and inversely proportional to the jet velocity. For a given chemistry, there may be only a small range of relevant parameter space that is accessible, even on terascale computers, with a given number of grid points. Although we are still in the process of parameter selection, we estimate the grid spacing to be 15 microns in each direction. These estimates are based on two- and three-dimensional turbulent simulation tests accounting for the local structure of highly strained, extinguishing flames, which require approximately three times more resolution than what would be estimated from steady extinction conditions. Our target grid number is 0.2 billion grid points, allowing a moderate Reynolds number.

For the compressible code, assuming uniform grid and that the acoustic CFL criterion controls the time-step, the cost for simulation of a single transient jet time is proportional to $N\frac{A}{M}N_H$ where N is the total grid number, M is the Mach number (the ratio of the jet velocity to the sound speed), N_H is the number of grid points resolving the jet height, and A is the factor by which the time step must be smaller than the acoustic CFL stability limit. Many such transient times must be run over the course of a given simulation. To reduce the overall computational cost the Mach number will be maximized while ensuring that the flow is essentially incompressible. Other parameters will be selected in order to maximize the Reynolds number for the given computational effort and to give the desired levels of extinction.

2.7 Formulation

The DNS solves a coupled system of time varying partial differential equations (PDEs) governing the conservation of mass, momentum, and energy, and species continuity. These governing equations are outlined in §2.7.1. The PDEs are supplemented with additional constitutive relationships, such as the ideal gas equation of state, and models for reaction rates, molecular transport, and thermodynamic properties.

2.7.1 Governing Equations

The equations governing reacting flows may be written in conservative form as

$$\frac{\partial \rho}{\partial t} = -\nabla_\beta \cdot (\rho \mathbf{u}_\beta), \tag{1}$$

$$\frac{\partial (\rho \mathbf{u}_\alpha)}{\partial t} = -\nabla_\beta \cdot (\rho \mathbf{u}_\alpha \mathbf{u}_\beta) + \nabla_\beta \cdot \boldsymbol{\tau}_{\beta\alpha} - \nabla_\alpha p + \rho \sum_{i=1}^{N_s} Y_i \mathbf{f}_{\alpha i}, \tag{2}$$

$$\begin{aligned} \frac{\partial (\rho e_0)}{\partial t} &= -\nabla_\beta \cdot [\mathbf{u}_\beta (\rho e_0 + p)] + \nabla_\beta \cdot (\tau_{\beta\alpha} \cdot \mathbf{u}_\alpha) \\ &\quad -\nabla_\beta \cdot \mathbf{q}_\beta + \rho \sum_{i=1}^{N_s} Y_i \mathbf{f}_{\alpha i} \cdot (\mathbf{V}_{\alpha i} + \mathbf{u}_\alpha), \end{aligned} \tag{3}$$

$$\frac{\partial (\rho Y_i)}{\partial t} = -\nabla_\beta \cdot (\rho Y_i \mathbf{u}_\beta) - \nabla_\beta \cdot (\rho Y_i \mathbf{V}_{\beta i}) + W_i \dot{\omega}_i, \tag{4}$$

where ∇_β is the gradient operator in direction β, Y_i is the mass fraction of species i, W_i is the molecular weight of species i, $\tau_{\beta\alpha}$ is the stress tensor, $\mathbf{f}_{\alpha i}$ is the body force on species i in direction α, $\mathbf{q}_\beta$ is the heat flux vector, $\mathbf{V}_{\beta j}$ is the species mass diffusion velocity, $\dot{\omega}_i$ is the molar production rate of species i, and e_0 is the specific total energy (internal energy plus kinetic energy),

$$e_0 = \frac{\mathbf{u}_\alpha \cdot \mathbf{u}_\alpha}{2} - \frac{p}{\rho} + h, \tag{5}$$

and h is the total enthalpy (sensible plus chemical). Throughout this document, α, β, γ will indicate spatial indices and i, j, will indicate species indices unless stated otherwise. Repeated spatial indices imply summation. For example, in cartesian coordinates,

$$\nabla_\beta \cdot (\rho \mathbf{u}_\beta) = \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} + \frac{\partial (\rho w)}{\partial z},$$

where u, v, and w are the velocity components in the x, y, and z directions, respectively. Only $(N_s - 1)$ species equations are solved because the sum of the N_s species equations yields the continuity equation. The mass fraction of the last species is determined from the constraint

$$\sum_{i=1}^{N_s} Y_i = 1. \tag{6}$$

Assuming an ideal gas mixture, the equation of state is given as

$$p = \frac{\rho R_u T}{W}, \tag{7}$$

where R_u is the universal gas constant and W is the mixture molecular weight given by

$$W = \left(\sum_{i=1}^{N_s} Y_i / W_i \right)^{-1} = \sum_{i=1}^{N_s} X_i W_i. \tag{8}$$

The species mass fractions (Y_i) and mole fractions (X_i) are related by

$$\frac{Y_i}{X_i} = \frac{W_i}{W}. \tag{9}$$

Relevant thermodynamic relationships between enthalpy and temperature for an ideal gas mixture include

$$h = \textstyle\sum_{i=1}^{N_s} Y_i h_i, \qquad h_i = h_i^0 + \int_{T_0}^{T} c_{p,i}\,\mathrm{d}T,$$

$$c_p = \textstyle\sum_{i=1}^{N_s} c_{p,i} Y_i, \qquad c_p - c_v = R_u/W.$$

where h_i is the enthalpy of species i, h_i^0 is the enthalpy of formation of species i at temperature T_0, and c_p and c_v are the isobaric and isochoric heat capacities, respectively.

2.7.2 Constitutive Relationships

The stress tensor, species diffusion velocities, and heat flux vector in equations (2)-(4) are given by [63, 72, 75]

$$\tau_{\beta\alpha} = \tau_{\alpha\beta} = \mu\left[\nabla_\alpha \mathbf{u}_\beta + \nabla_\beta \mathbf{u}_\alpha\right] - \delta_{\alpha\beta}\left(\frac{2}{3}\mu - \kappa\right)\nabla_\gamma \cdot \mathbf{u}_\gamma, \tag{10}$$

$$\mathbf{V}_{\alpha i} = \frac{1}{X_i}\sum_{j=1}^{N_s}\frac{Y_j}{X_j} D_{ij}\mathbf{d}_{\alpha j} - \frac{D_i^T}{\rho Y_i}\nabla_\alpha(\ln T), \tag{11}$$

$$\mathbf{q}_\alpha = -\lambda\nabla_\alpha T + \sum_{i=1}^{N_s} h_i \mathbf{J}_{\alpha i} - \sum_{i=1}^{N_s}\frac{p}{\rho Y_i} D_i^T \mathbf{d}_{\alpha j}. \tag{12}$$

where μ is the mixture viscosity, κ is the bulk viscosity, D_{ij} are the *multicomponent* diffusion coefficients, D_i^T is the thermal diffusion coefficient for species i, λ is the thermal conductivity, $\mathbf{J}_{\alpha i} = \rho Y_i \mathbf{V}_{\alpha i}$ is the species diffusive flux, and $\mathbf{d}_{\alpha i}$ is the diffusion driving force for species i in direction α, given by [63, 72, 75]

$$\mathbf{d}_{\alpha i} = \underbrace{\nabla_\alpha X_i}_{1} + \underbrace{(X_i - Y_i)\nabla_\alpha(\ln p)}_{2} + \underbrace{\frac{\rho Y_i}{p}\left[\mathbf{f}_{\alpha i} - \sum_{j=1}^{N_s} Y_j \mathbf{f}_{\alpha j}\right]}_{3}. \tag{13}$$

The driving force vector involves thermodynamic forces generated by gradients in concentration (term 1), gradients in pressure (term 2) also called "barodiffusion," and due to any body force such as an electrical or gravitational field (term 3). Equation (13) allows for the possibility that the force on each species, $\mathbf{f}_{\alpha i}$, is different, though in the case of a gravitational field, $\mathbf{f}_{\alpha j} = \mathbf{g}_\alpha$, and term 3 is identically zero. In the following sections, we will consider the fluxes given in equations (10)-(12) in more detail.

2.7.3 Stress Tensor

For monatomic gases, κ is identically zero, and it is often neglected for polyatomic gases as well [63, 72, 75]. It will be neglected in all discussion herein. This simplifies (10) to

$$\tau_{\beta\alpha} = \tau_{\alpha\beta} = \mu \left(\nabla_\alpha \mathbf{u}_\beta + \nabla_\beta \mathbf{u}_\alpha - \frac{2}{3}\delta_{\alpha\beta} \nabla_\gamma \cdot \mathbf{u}_\gamma \right), \tag{14}$$

which is the form that will be used for this work.

2.7.4 Mass Diffusion Flux

It should be noted that all diagonal components of the multicomponent diffusion coefficient matrix (D_{ii}) are identically zero [75]. Also, the diffusive fluxes and driving forces for all species must sum to zero,

$$\sum_{i=1}^{N_s} \mathbf{J}_{\alpha i} = \sum_{i=1}^{N_s} \rho Y_i \mathbf{V}_{\alpha i} = 0, \qquad \sum_{i=1}^{N_s} \mathbf{d}_{\alpha i} = 0. \tag{15}$$

Equation (11) is often approximated as [63, 72, 75, 103]

$$\mathbf{V}_{\alpha i} = -\frac{D_i^{\text{mix}}}{X_i} \mathbf{d}_{\alpha i} - \frac{D_i^T}{\rho Y_i} \nabla_\alpha \ln T, \tag{16}$$

where D_i^{mix} are "mixture-averaged" diffusion coefficients given in terms of the binary diffusion coefficients ($\mathcal{D}_{ij}$) and the mixture composition as

$$D_i^{\text{mix}} = \frac{1 - X_i}{\sum_{j \neq i} X_j / \mathcal{D}_{ij}}, \tag{17}$$

where the binary coefficient matrix is symmetric ($\mathcal{D}_{ij} = \mathcal{D}_{ji}$), and the diagonal elements are zero ($\mathcal{D}_{ii} = 0$). If we assume that body forces act in the same manner on all species and baro-diffusion (term 2 in (13)) is negligible, then (13) becomes $\mathbf{d}_{\alpha i} = \nabla_\alpha X_i$. If we further neglect the Soret effect, (the second term in (11) and (16)), then (16) reduces to

$$\mathbf{V}_{\alpha i} = -\frac{D_i^{\text{mix}}}{X_i} \nabla_\alpha X_i, \tag{18}$$

which, using (9), can be expressed in terms of mass fractions as

$$\begin{aligned} \mathbf{V}_{\alpha i} &= -\frac{D_i^{\text{mix}}}{Y_i} \left[\nabla_\alpha Y_i + \frac{Y_i}{W} \nabla_\alpha W \right] \\ &= -\frac{D_i^{\text{mix}}}{Y_i} \left[\nabla_\alpha Y_i - Y_i W \sum_{j=1}^{N_s} \frac{\nabla_\alpha Y_j}{W_j} \right]. \end{aligned} \tag{19}$$

Studies on the effects of thermal diffusion suggest that the Soret effect is much more important for premixed flames than for nonpremixed flames, the Dufour effect is of little importance in either premixed or nonpremixed flames [70].

2.7.5 Heat Flux

The heat flux vector is comprised of three components representing the diffusion of heat due to temperature gradients, the diffusion of heat due to mass diffusion, and the Dufour effect [68, 70, 72, 75, 103]. In most combustion simulations, the Dufour effect is neglected, and (12) may be written as

$$\mathbf{q}_\alpha = -\lambda \nabla_\alpha T + \sum_{i=1}^{N_s} h_i \mathbf{J}_{\alpha i}. \tag{20}$$

All treatment here will be restricted to adiabatic systems. While it is certainly true that radiation and other heat-loss mechanisms are important in many combustion applications, this complication will not be considered here.

2.8 Code Overview

The structure and flow of the code S3D is described in this section and illustrated in Figure 3. The S3D code is structured to either execute in the run mode or postprocessing mode. In the run mode, the code integrates the governing equations forward in time based on a case specific initialization of the primitive variables. In this mode, all required operations are directed by the routine solve-driver. In postprocessing mode the code executes with the same processor topology as in the run mode but all required operations are directed by the routine post-driver.

After the initialization of the primitive variables for each time step the convective, diffusive and chemical terms in the conservation equations are updated, once for each of the six stages of the fourth-order accurate explicit Runge-Kutta time advancement solver. The main kernels in this solver where over 95% of the computation occurs are given below:

- Chemistry - Computes chemical reaction rate source terms for species equations. The chemical kinetics data is preprocessed and the code to compute the reaction rates, named as "getrates", is generated by the Chemkin compatible preprocessing utility Autogetrates package . The routines are packaged in a separate module which acts as an interface to the code and abstracts the actual implementation of the reaction rates computation. This will allow us to use different versions of the getrates subroutine targeted at different platforms.

- Transport - Computes molecular transport properties for the species. The properties computed include the viscosity, thermal diffusivity and species mass diffusivities. The code is linked with the transport library which is part of the standard Chemkin suite.

- Thermodynamics - Computes the thermodynamic properties such as enthalpy and specific heats of the mixture. The thermodynamic data are given in the Chemkin compatible format and are preprocessed through the

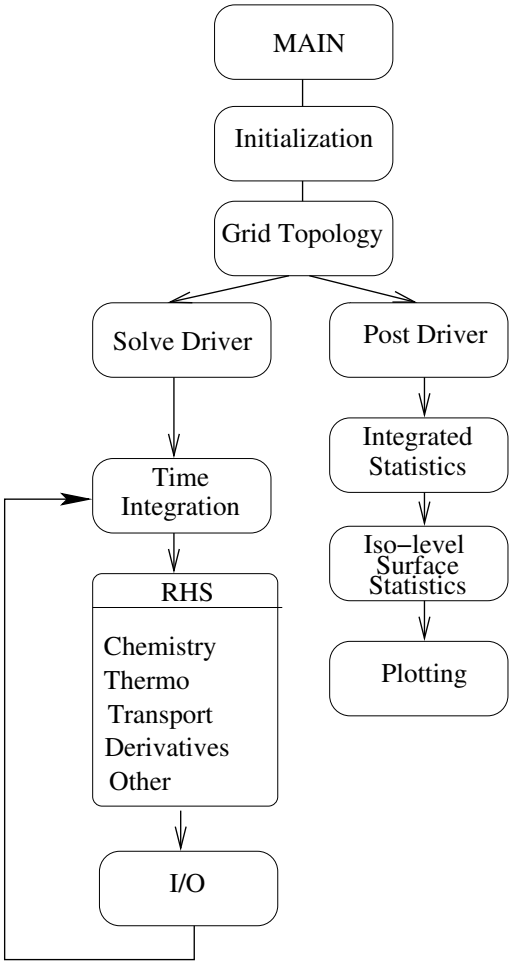

Figure 3: Program flow diagram for S3D

chemkin interpreter. Rather than directly evaluate the properties using the chemkin routines, the code employs a tabulation and lookup strategy.

- Derivatives - Computes the spatial derivatives of the primitive and conserved variables using higher order finite difference operators. The code uses non-blocking sends and receives to exchange the data at the processor boundaries among different processors.

- Other RHS - The right hand side of the time advance equation involves all of the above mentioned operations and the convection terms. These terms are summed up according to the governing equations. All operations involved in this procedure are lumped into the Other RHS module for accounting purposes.

- Time Integration - Advances the solution in time using a 4th order accurate Runge-Kutta scheme. This module also includes an error controller which routinely checks for the time accuracy of the solution and adjusts the time step to achieve the desired error tolerances.

2.9 S3D Software Performance and Improvements

The scientific benefit of the INCITE DNS calculations will be maximized when the code is tuned to require the least amount of computational time per step and grid point. Within the fixed INCITE allocation, this could allow an increased grid number and/or a longer physical simulation time. Such increases would help to achieve the scientific goals by allowing higher Reynolds numbers, a greater sample of turbulent structures from which to take statistics, and/or a more complete temporal development of the turbulent flame. Therefore, the INCITE team, together with NERSC consultant David Skinner, are working towards understanding and improving the performance of S3D on the Seaborg IBM SP platform. Considering that the code has been run on this platform for several years, improvements are expected to be incremental. Also, the INCITE team has been working on porting the code to the Pheonix Cray X1 architecture at ORNL. Due to the substantially different vector architecture, much more significant gains have already been achieved, and further gains may be forthcoming.

The performance of the computational implementation of the DNS software can be measured in terms of the following metrics: (i) Computational time required for a given problem size or larger problem size for a given computational effort (ii) Communication overhead and scaling of a parallel computation over several hundred to several thousand processors (iii) The maximum problem size that can fit onto a machine given the system memory limitations. Performance evaluation and improvements have been divided into the following areas:

- scalar performance evaluation and improvements

- evaluation of parallel scaling and communication overheads

- evaluation of memory limitations

2.10 Test problem description

For the scalar profiling and the parallel scaling studies a pressure wave problem on 40^3 grid points per processor was chosen for simulation. The tests were conducted using detailed CO-H_2 chemistry. The choice of problem size and chemical complexity is representative of the work load associated with the INCITE run, although slightly smaller. The initial condition consists of a gaussian temperature profile centered in the domain with periodic boundary conditions. When integrated in time, the initial temperature non-uniformity gives rise to pressure waves and spreading of the temperature profile.

2.11 Scalar profiling and performance improvements

The scalar performance was measured by evaluating the computational time per time step and grid point. The most CPU intensive sections of the code were identified by profiling the code execution on various platforms including Linux Clusters, IBM SP, and CrayX1. Rather than describe exhaustively the code by subroutine, these sections were grouped according to a modular physics based decomposition of the computation as described in Section 2.8.

2.11.1 Performance on Seaborg

The profiling on Seaborg was done using "Xprofiler" which is a GUI based performance profiling tool distributed as part of the IBM Parallel Environment for AIX. It was used to graphically identify the functions which are the most CPU intensive in S3D. It provides results in the form of a call tree as well as a flat profile that details the time spent in each routine. The results of the profiling tool are analyzed and the time spent in each of the subroutines is lumped into one of the several modules described earlier. The profiling results of the original code are shown in Fig. 4. The figure shows a breakdown of the time spent per time-step and grid point for the code evaluated from scalar runs on Seaborg at NERSC. As expected, the code spends most of its time in the chemistry, transport and thermodynamic modules, in that order. Changes were implemented in these three modules as described below.

1. Most sections of the code use non-dimensional form of the variables and equations to minimize the truncation error. However the transport and chemistry modules are interfaced with the CHEMKIN libraries, which use dimensional quantities in order to be able to use the standard chemical and transport properties databases. The profile showed that a considerable amount of time was being spent by the code in converting the relevant variables between dimensional and non-dimensional units. Several sections of the code were rewritten to minimize the unit conversions and reuse some of the converted data when available. This resulted in around 8% improvement in performance.

2. The computation of chemistry and transport properties involved calls to the exponential and logarithm mathematical functions. To minimize the

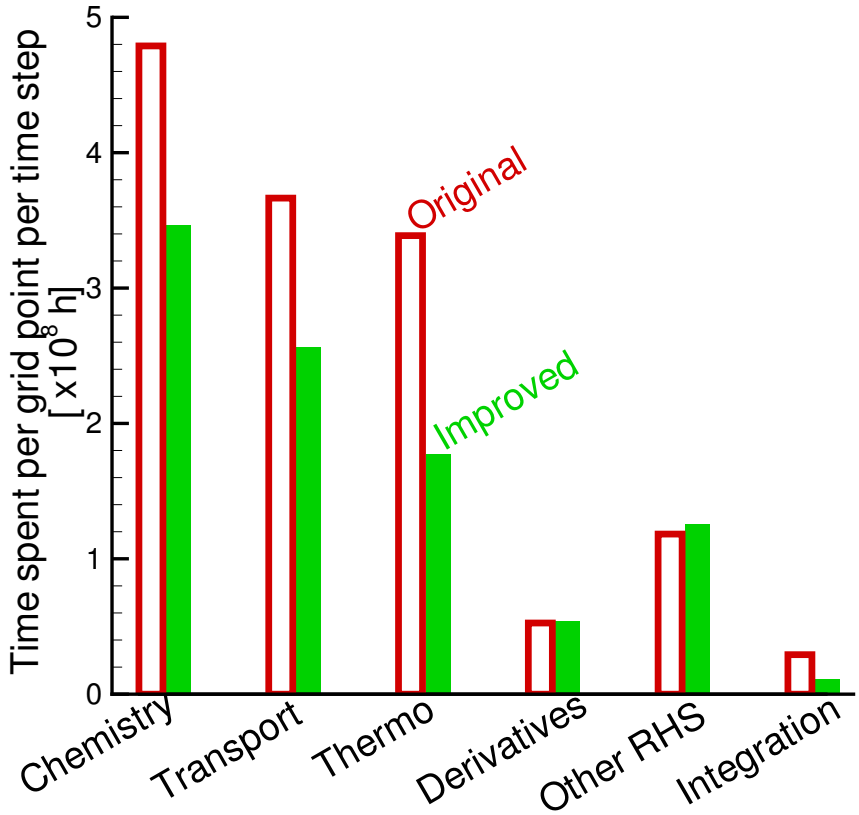

Figure 4: Performance improvements on Seaborg IBM SP at NERSC

cost of computing these mathematical functions the code was linked to an accelerated math library written in POWER3 assembly code that is available on Seaborg. The Mathematical Acceleration SubSystem (MASS) consists of libraries of tuned mathematical intrinsic functions, which offer improved performance over the standard mathematical library routines at the expense of not being as accurate in some cases. The use of these libraries led to a performance improvement of around 10%.

3. The evaluation of the thermodynamic properties involves evaluating polynomials of up to 7th degree in temperature and are very expensive to calculate. Therefore the current thermodynamics module tabulates most of these properties as a function of the temperature. Extracting the properties from this table instead of computing them has proved to be an effective strategy in minimizing the cost of computation. However, only one property, namely the Gibbs energy, continued to be computed instead of tabulated. This property is used in computing the equilibrium constants of reversible reactions in the chemistry module. In the improved version of the code the Gibbs energy of each species is tabulated as a function of temperature, like other thermodynamic properties. The tabulation strategy led to savings in CPU time of around 8%.

The profiling results obtained from the improved version of the code are shown in fig. 4. After these improvements, the code as a whole spends 26% less time on a single processor run. The scalar computing cost was lowered from $1.5\text{x}10^{-7}$hours/gridpoint/timestep to $1.1\text{x}10^{-7}$ hours/gridpoint/timestep.

Future scalar performance improvements on Seaborg may be possible. These will focus firstly on the evaluation of reaction rates and transport properties. The use of vectorized exponential functions in the MASS library may lead to further gains in the reaction rate evaluation. In transport property evaluations, reorganization of loops in legacy code may lead to more effective compiler optimizations. Using the xprofiler tool, several instances of unnecessary repetition of dynamic allocation of temporary data-structures in chemistry and thermo kernels have been identified. Elimination of these may result in small gains.

2.11.2 Porting the code to Cray X1

The profiling on Phoenix was done using CrayPat. The CrayPat suite of tools do not require the code to be recompiled in order to perform the profiling. The "pat_build" utility is used to instrument the compiled executable with tracing and polling calls and produce a modified executable. The report produced on execution is analyzed using the "pat_report" utility. The improved version obtained on Seaborg as a result of the optimization exercise is used as the starting point for the CRAY experiments. The scalar execution time on Cray was 1.8e-7 hrs/gridpoint/timestep. The profile of executing this code is shown in Fig. 5. It it seen that the code spends a disproportionately long time in the chemistry and thermodynamic modules. These modules were recompiled with the -rm option to obtain a detailed listing file that shows the optimizations

performed by the compiler and the sections of code that it was not able to optimize. It was found that the compiler was not able to vectorize the chemistry module. Hence a large portion of the time was spent in evaluating the scalar version of the exponent function. Similarly the thermodynamic module had several functions that involved type constructs and allocatable data objects. The Cray compiler was not able to inline these functions and as a result their callers were not being vectorized.

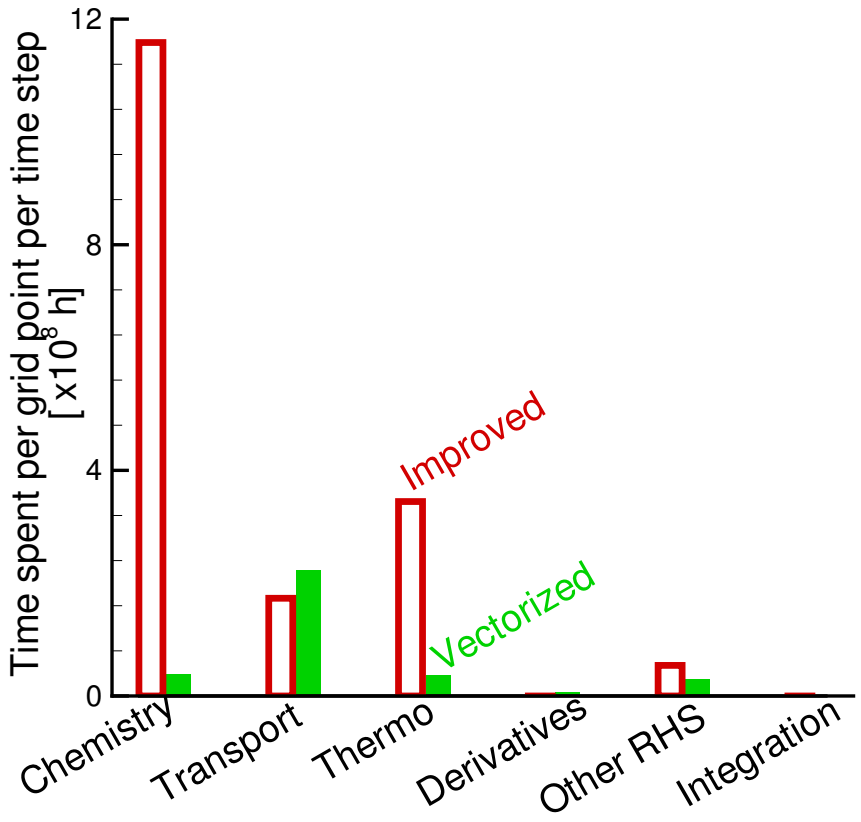

Figure 5: Performance improvements on Pheonix Cray X1 at ORNL

The problem was rectified by rewriting a significant part of the chemistry module in a form suitable for vectorization. In particular, instead of computing the reaction rates at every grid point separately, a new routine was written to compute the reaction rates all over the domain. This modification made it possible for the compiler to invoke the vector exponent which resulted in a significant improvement. The issue with the thermchem module was resolved by replacing some of the type constructs and allocatable arrays with generic typed variables and static arrays. Furthermore some parts of the code were manually

inlined to assist the vectorization.

As a result of these modifications the compute cost went down to 3.55×10^{-8} hours/gridpt/timestep. As seen in Fig. 5 the cost of computing the chemistry and thermodynamics modules are reduced to insignificant levels as a result of the vectorization.

Further improvements on the Cray X1 architecture are likely. The primary candidate is the evaluation of transport properties, which does not vectorize well due to the present structure of loops.

2.12 Evaluation of parallel scaling and communication overheads

S3D is a mature DNS code and has been demonstrated to scale very well up to 4000 IBM SP processors. Also, the inter-process communications are minimal and exist only between the nearest neighbors in the processor domain topology. Hence, communication was expected not to be a bottleneck for the performance of the code.

2.12.1 Scaling on Seaborg

A series of comparison runs of S3D on 1, 8, 64, 256, and 512 processors of increasing total problem size proportional to the number of processors were done on Seaborg. This comparison was between the S3D code as it started out on Seaborg and the code as of Q1 2005. The Integrated Performance Monitoring (IPM) tool was deployed to analyze the communication overhead and scaling performance. IPM is a portable profiling infrastructure which provides users with a concise report on the execution of parallel jobs. The IPM reports, generated by David Skinner, are available at: `http://www.nersc.gov/~dskinner/tmp_s3d/`. The files are named s3d_orig_N or s3d_inciteQ1_N, where "orig" refers to the original code, "inciteQ1" refers to the new code, and N is the number of tasks for the particular run.

A great deal of information is contained in the IPM reports. The main points are summarized as below.

- The code is scaling well. Figure 6 shows the total wall-clock time versus the number of tasks. Aside from one outlier at 256 tasks the new code shows consistently better performance approaching 14%.
- The code scaling is not communication bound showing only 10-20% communication time.
- The communication topology looks to be well blocked as seen in the lower part of the IPM reports. There may be a more effective task ordering that could lead to more SMP vs. switch traffic, but gains are not expected to be significant.

- The amount of time spent in MPI_Barrier is sometimes an appreciable fraction of the communication time. This suggests that load balance may be improved to some small extent. The first step toward this has been completed, namely removing most of the unnecessary barriers. While this does not improve performance directly it does expose the tasks which are blocked. Discussions have been started about how the layout of processes assigned to Seaborg nodes might be optimized.

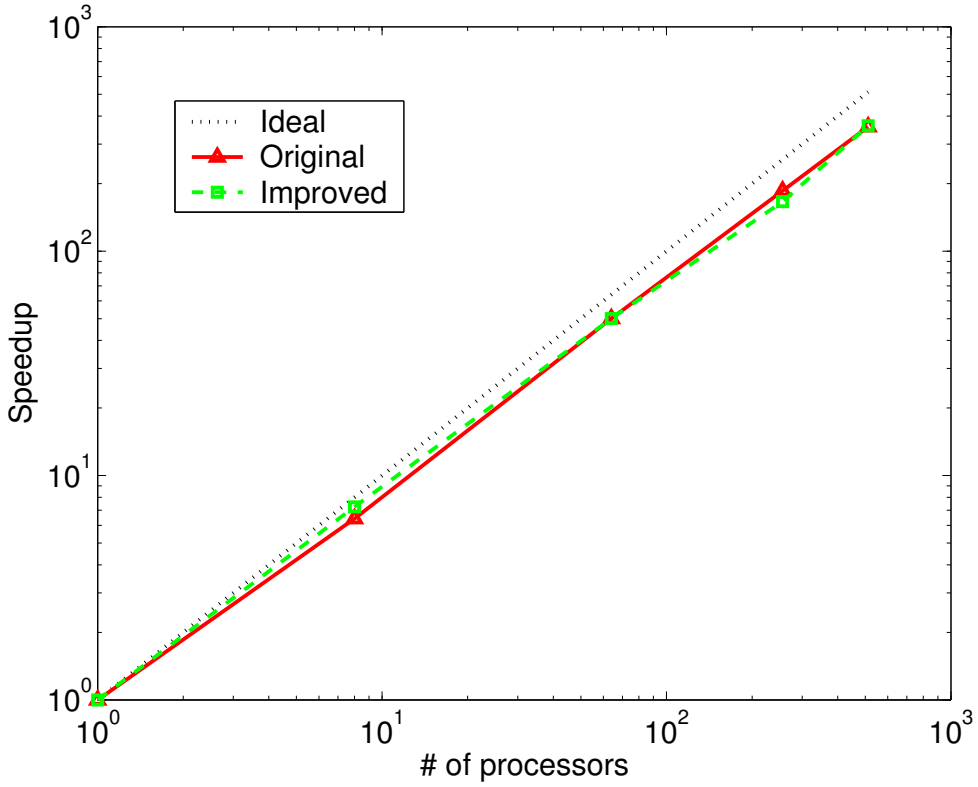

Figure 6: Code scaling on IBM SP

2.12.2 Scaling on Cray X1

Preliminary scaling tests on 1,8,64, and 256 processors have been performed on the Cray X1. Figure 7 shows the speed-up of the code against the number of processors. As in the Seaborg tests, the total problem size is increased proportional with the number of processors. The code is also scaling reasonably well on this platform. The appearance of better scaling of the "improved" code is due to the higher ratio of computation to communication per processor. Further

work is needed to evaluate whether any scaling improvements can be obtained on this architecture.

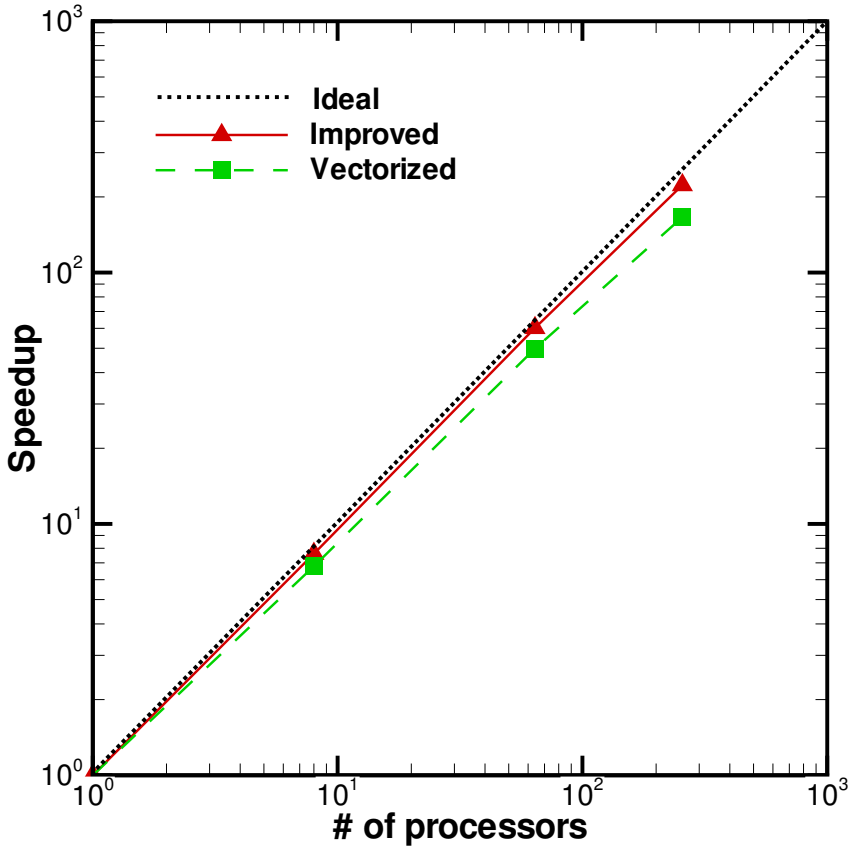

Figure 7: Code scaling on Cray X1

2.13 Evaluation of memory limitations

Due to the large size of the INCITE calculation, it is necessary to allow a larger problem size per task than usual. On Seaborg, it was initially found that the problem size was memory limited to approximately 90000 grid points per task. However, this difficulty was quickly remedied by utilizing the appropriate compiler flag, -bmaxdata, allowing the use of up to nearly 3 million grid points per task.

In order to decide the precise parameter space for the INCITE calculation, it is necessary to run many smaller test calculations, which are in themselves computationally very demanding. The INCITE team is employing a new local CRF Opteron Linux cluster with Infiniband switches for the test calculations. On this machine, which has a very high processor speed, the code was found to be memory limited. The memory usage was analyzed using a tool named Valgrind. Based on the results of the analysis, several unnecessary arrays were eliminated and different sections of the code were made to share the same memory space for their operations. Such improvements reduced the memory footprint of the code by approximately 25-30 %, allowing a equivalent increase in the size, and therefore relevance, of the test problems.

2.14 Q3 Progress and Dynamics

2.14.1 Performance improvements on Seaborg

- As before, the code execution was profiled using 'Xprofiler' and it was noticed that a large number of calls were being made to 'malloc' and 'free', which are the functions used for dynamic allocation and deallocation respectively. These calls constituted around 12% of the total running time and were worth investigating. Since S3D does not invoke such a large number of dynamic memory allocation calls at each time step, the source of these calls was puzzling. Upon investigation, it was found that wherever a non-contiguous section of a multi-dimensional array is passed as an argument to a function, a small temporary array is created through malloc and then deallocated after that call was over. In some parts of the code, such function calls were inside a loop that gets repeated several hundred thousand times every time step. Instead of allocating this array space once, the compiler places several calls to malloc and free. The issue was resolved by manually creating a small temporary array for storing the non-contiguous data and passing it to the function.

- Upon analyzing the results of several test problems in preparation for the main INCITE calculation, it was found that the species H_2O_2 has a negligible role in the non-premixed flame dynamics at the conditions chosen for this study. Therefore, the chemical model was reduced from 12 species to 11 species by removing H_2O_2 in order to trim the chemistry evaluation cost further.

- The derivative routines use MPI calls for communicating the boundary data with the neighbors. Whenever the data to be communicated is not contiguous, MPI derived data types are used to perform the data transfer. Based on the experience on other platforms, such as the Opteron cluster at Sandia, the derived data types were retired from the prominent functions and the communication is now done by packing the data together in a second array and then communicated using non-blocking MPI routines.

As a result of the aforementioned changes, the scalar execution cost of the code has been streamlined to 0.8×10^{-7} hours/gridpoint/timestep. Note that the cost was 1.5×10^{-7} hours/grid point/timestep at the beginning of FY2005, and then reduced to 1.1×10^{-7} hours/grid point/timestep at the end of Q2. The improvements made in FY2005 to date have improved the efficiency of the code by 45%.

2.14.2 Performance improvements on Cray X1

In addition to carrying over the improvements mentioned above, further enhancements were made specifically targeted at the Cray X1 platform.

- It was found that the Cray compiler was not able to inline some of the functions in the transport module. This affected its ability to vectorize those sections of the code. To correct this problem, the relevant subroutines were manually inlined to enable vectorization.

- In the transport module, several nested loops were present in which the inner loop has dependency on the outer loop. Since some of the transport property matrices are symmetric by definition, the code takes advantage of this property and performs fewer floating point operations by using such a complex inner loop. While this approach is advantageous on other platforms, it is not conducive to vectorization on the Cray X1. Therefore, a modified version of code was written for the X1, in which the entire matrix is evaluated in spite of its symmetry.

- There were other instances of nested loops in the transport module which were not being fully vectorized due to the dependence of a portion of the inner loop calculations on the outer loop. This was resolved by fissioning the loop into two different loops such that the dependence was removed.

- The Cray systems have a capability known as co-arrays which are a syntactic extension to the fortran language, better suited for the purposes of data exchange between processes. Compared to the typical message passing through MPI, co-array syntax makes the program suitable for analysis and optimization by the compiler. This promises greater opportunity for minimizing the data transfer latency. S3D was modified to add the co-array capability, wherein communications can be performed using the co-array syntax instead of MPI calls. It was found that co-array communication improves the performance greatly compared to the MPI communication made using derived data types. However, when the data is packed into a contiguous array and then communicated using MPI calls without using any derived data types, similar improvements were observed. It seems that the co-arrays can provide a cleaner interface to data exchange without the overheads of MPI communication using derived data types.

As a result of the aforementioned changes, the scalar execution cost of the code is currently down to 1.5×10^{-8} hours/gridpoint/timestep. Note that the cost was 1.8×10^{-7} hours/grid point/timestep at the beginning of FY2005, and then down to 3.5×10^{-8} hours/grid point/timestep at the end of Q2. The improvements made in FY2005 to date have improved the efficiency of the code almost ten-fold.

2.15 Scientific Insights from DNS Simulations

In preparation for the large INCITE calculation, several smaller size production runs have been performed in the same physical configuration. A 40 million node calculation was performed on the MPP2 Linux cluster at PNNL, the details of which are discussed in section 2.5. In addition a second production calculation using 100 million grid points was performed on the IBM Seaborg at NERSC and the Cray X1E at ORNL. Relative to these runs, we expect to increase the Reynolds number, domain size and the amount of extinction for the INCITE calculation. The following paragraphs briefly discuss the scientific insights gained from postprocessing the data obtained from the PNNL run.

2.15.1 Mixing timescales

Models for molecular mixing are required in many approaches for the simulation of turbulent combustion. In particular, molecular mixing is the central modeling question in the PDF approach [105]. A key element in the PDF approach is a mixing time scale. Pope [106] has recently pointed out that model predictions are dependent on the choice of the timescale, and different choices are appropriate for different problems. Normally, the timescale is assumed to be the same for each different scalar, and the same order of magnitude as the large scale turbulence timescale. In flames, differential diffusion and the strong interplay with mixing and reaction might degrade these assumptions. It is difficult to directly assess these assumptions in a-posteriori tests, and measurements of reacting scalar mixing are not yet possible. DNS of reacting flows with detailed chemistry provides unique opportunities to evaluate such assumptions.

2.15.2 Mathematical definition

Here, a mixing timescale is defined for a scalar ϕ:

$$\tau_\phi = \frac{\overline{\phi'^2}}{\overline{\chi_\phi}}, \tag{21}$$

where a dissipation rate χ_ϕ is defined for the scalar as:

$$\chi_\phi = 2D_\phi \nabla\phi \cdot \nabla\phi, \tag{22}$$

and D_ϕ is the mixture-averaged diffusion coefficient for the scalar. A mechanical timescale, which represents the characteristic large-scale turbulence timescale is

defined by:

$$\tau_u = \frac{\overline{k}}{\overline{\epsilon}}, \tag{23}$$

where k is the turbulence kinetic energy and ϵ is its dissipation rate. The average $\overline{(\ldots)}$ in these simulations is taken over the spanwise and streamwise directions, which are statistically homogeneous. The ratio of the scalar to mechanical mixing timescale is usually assumed to be order unity. Here a timescale ratio is defined as:

$$r_\phi = \frac{\int_0^{\delta_Z} \tau_u dy}{\int_0^{\delta_Z} \tau_\phi dy}, \tag{24}$$

where δ_Z is the y location at which $Z < 0.05$. Integration across the y direction allows convenient presentation of the results as a single timescale ratio, and reduces statistical scatter.

2.15.3 Effect of Schmidt Number

Figure 8 shows r_Z, the ratio of the mechanical timescale to that of the mixture fraction versus the simulation time normalized by the transient jet time U/H. The mixture fraction is a passive scalar in that its transport equation contains no reactive source term. A Lewis number of unity is assumed to calculate the diffusivity. The figure shows that the timescale ratio is order unity throughout the simulation, varying between values of 2.1 to 1.0. The present result is similar

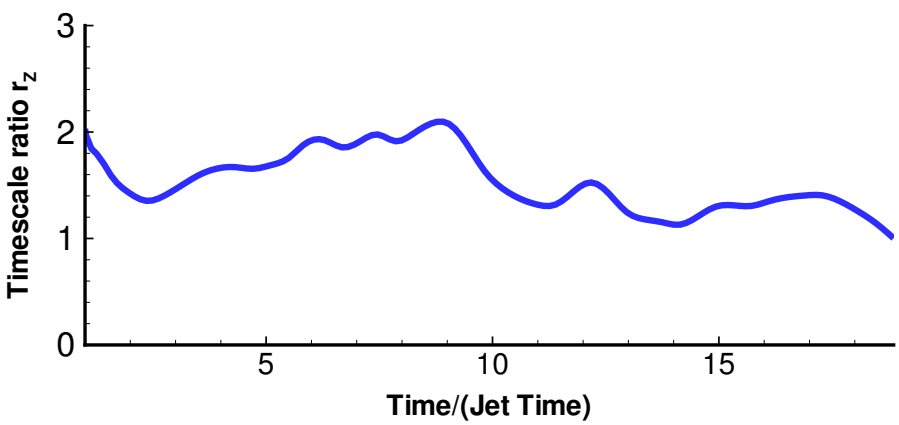

Figure 8: Mechanical to scalar timescale ratio for mixture fraction versus time.

to values reported by experiments, simple chemistry DNS and used successfully in modelling [108]. The result provides confirmation that the simple production equals dissipation assumption that r_Z should be order unity is valid for passive scalars with Schmidt numbers order unity, even in a reacting flow.

Scalars with non-unity Schmidt numbers can potentially have different mixing timescales. Figure 9 shows the mixing timescale ratio for the H, H_2 and

CO_2 mass fractions versus time, along with the passive scalar timescale. These different molecules have different diffusivities - H is the most diffusive while CO_2 is the least. The figure clearly shows there is an effect of diffusivity on the mixing timescales. The most diffusive scalar H has a maximum timescale ratio of 6.2 while CO2 has a maximum timescale ratio of 1.6. The difference between these values however is less than the difference in the diffusivities. The results indicate that differential diffusion effects may need to be incorporated within mixing models, at least at moderate Reynolds numbers. This conclusion could possibly be Reynolds number dependent. A parametric study in Reynolds will be required to determine any such dependence.

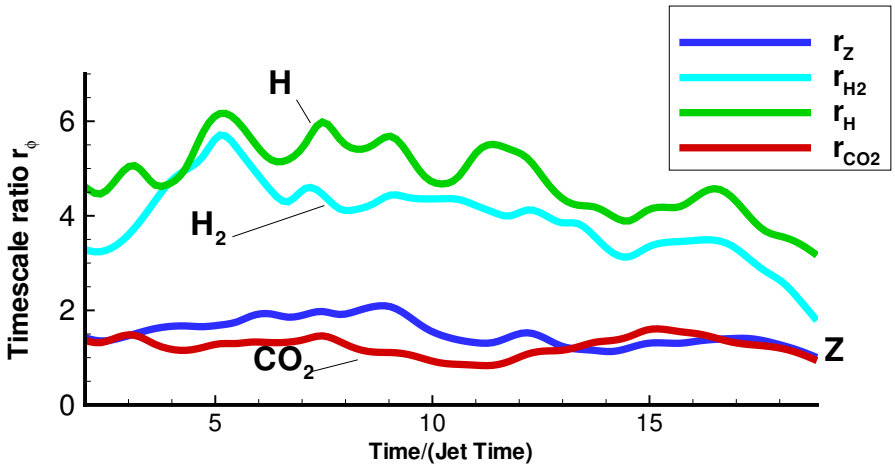

Figure 9: Mechanical to scalar timescale ratio for H, H_2 and CO_2 mass fractions and mixture fraction versus time.

2.15.4 Effect of Reaction-Diffusion coupling

The strong interplay between reaction and diffusion in nonpremixed flames can also affect mixing timescales. Figure 10 shows the mixing timescale ratios for HO_2, H_2O_2, and O and OH. Initially these timescales are ordered according to the diffusivity, but during the middle of the simulation, the timescales of O and OH increase and those of HO_2 and H_2O_2 decrease. This is a result of an interesting interplay between reaction and diffusion. Figure 11a) shows the HO_2 mass fraction, and Figure 11b) shows the OH mass fraction on a color scale for the time 10 H/U. In both figures, white contours of the scalar dissipation χ are overlaid. It may be observed that OH levels are lower in regions of high χ, while HO_2 levels are higher. In these high dissipation regions, conditions approach extinction and OH and O radicals are destroyed while the stable intermediates HO_2 and H_2O_2 are produced. This has a direct effect on the dissipation fields. Figure 12 shows the fields χ_{HO_2}, χ_{OH} and χ. As a result of

the strong production of HO_2 in high dissipation regions the χ_{HO2} and χ fields are coincident, while in these regions high OH dissipation does not occur. These high dissipation regions are most prevalent in the middle of the simulation, and together with these chemical effects lead to longer mixing timescales in the case of OH and shorter in the case of HO_2. Later, dissipation rates relax and O and OH return, leading to the increase of the timescales. These findings underline the importance of considering the interplay of diffusion and reaction, particularly when strong finite chemistry effects are involved. Preliminary results shown in figure 13 obtained from the calculations performed on the Cray X1 and X1E show similar trends as reported above for the PNNL data.

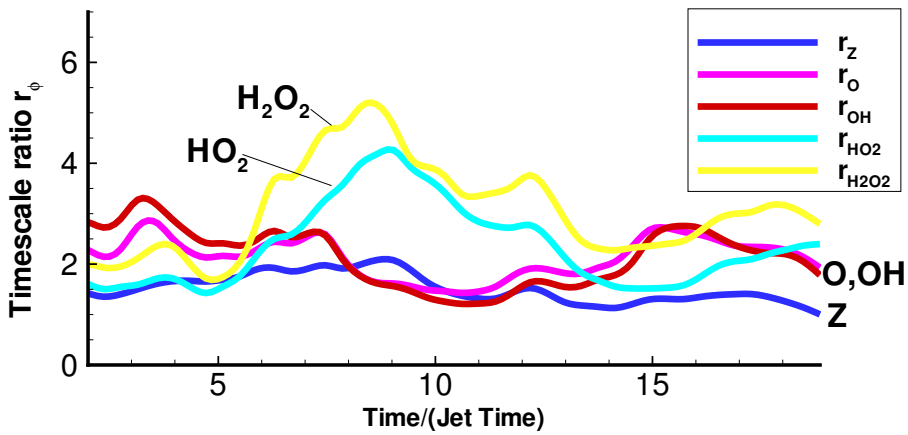

Figure 10: Mechanical to scalar timescale ratio for HO_2, H_2O_2, O and OH mass fractions and mixture fraction versus time.

It is worth noting that at present it is impossible to obtain this type of information any other way than by using the type of highly resolved simulation performed here. Experimental measurements of the scalar dissipation in flames are very difficult and even point-wise measurements have only recently become possible [107]. Full access to the 3D spatial and temporally resolved dissipation fields without the interference of noise, and of reacting scalars occurring in even thinner layers than those corresponding to mixture fraction is at present unthinkable experimentally.

2.16 Knowledge discovery from terascale simulated combustion data

Knowledge discovery from terascale datasets is a daunting task due to their sheer size and the complexity of the studied phenomena. For example, this INCITE calculation will produce 5 terabytes of data, vast in spatial, temporal and variable domains, creating a formidable challenge for subsequent analysis and interpretation; challenges also common to other Office of Science applications.

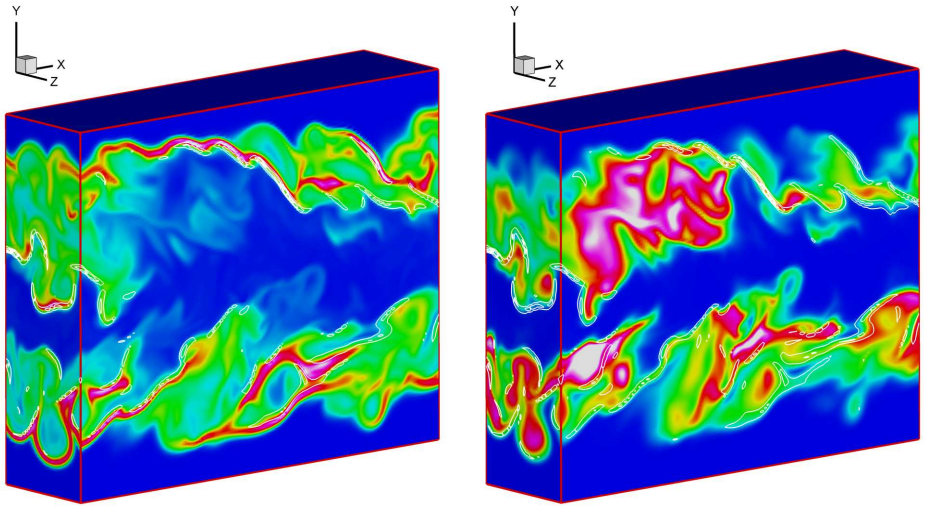

Figure 11: a) HO2 mass fraction on a color scale b) OH mass fraction on a color scale. White contours of χ.

Manipulating 5 Terabytes of raw data, and more in the future, will undoubtedly stress the network and the storage infrastructure. Moreover, knowledge-extraction is also compounded by the sheer complexity of the turbulent flow-fields, the phenomena being studied, and the existence of heterogeneous data types.

In order to meet these challenges, it would be desirable to have an integrated framework for data discovery that incorporates new intelligent interactive feature detection and tracking algorithms, innovative interactive parallel volume rendering techniques, and scalable data-sharing capabilities between platforms to provide a unified end-to-end solution for knowledge extraction from high-fidelity computational combustion simulations. In the absence of such a paradigm, we have painstakingly moved data from preliminary INCITE runs generated on NERSC and ORNL to a local Opteron Cluster at Sandia and analyzed the data using parallel postprocessing software already built into the S3D code. For example, obtaining and tracking volume averages and pointwise scalar statistics is relatively simple, whereas computing spatial correlations has proven to be more difficult.

2.17 Prospects for future combustion simulations on petascale platforms - breakthrough science and what is required

The computational cost of a DNS is largely determined by the spatial and temporal resolution requirements. The resolution requirements in a reactive

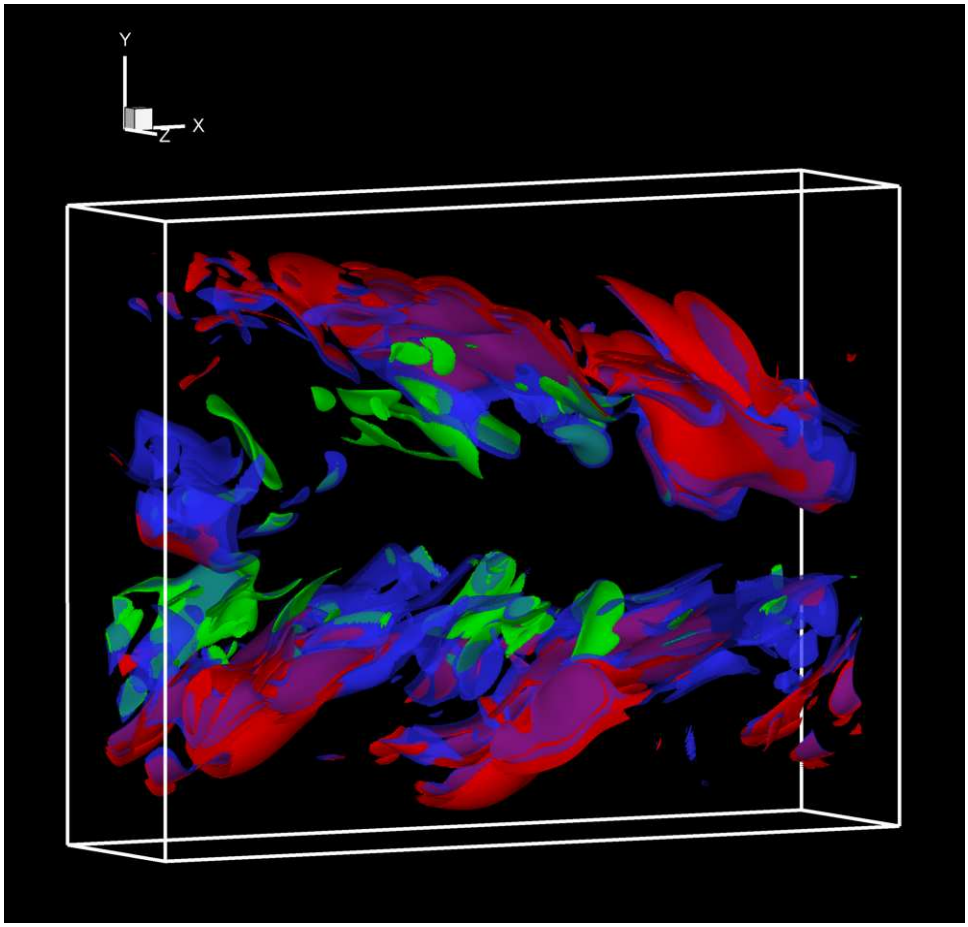

Figure 12: Iso-surfaces of dissipation rates. Blue: χ. Red: χ_{HO2}. Green: χ_{OH}.

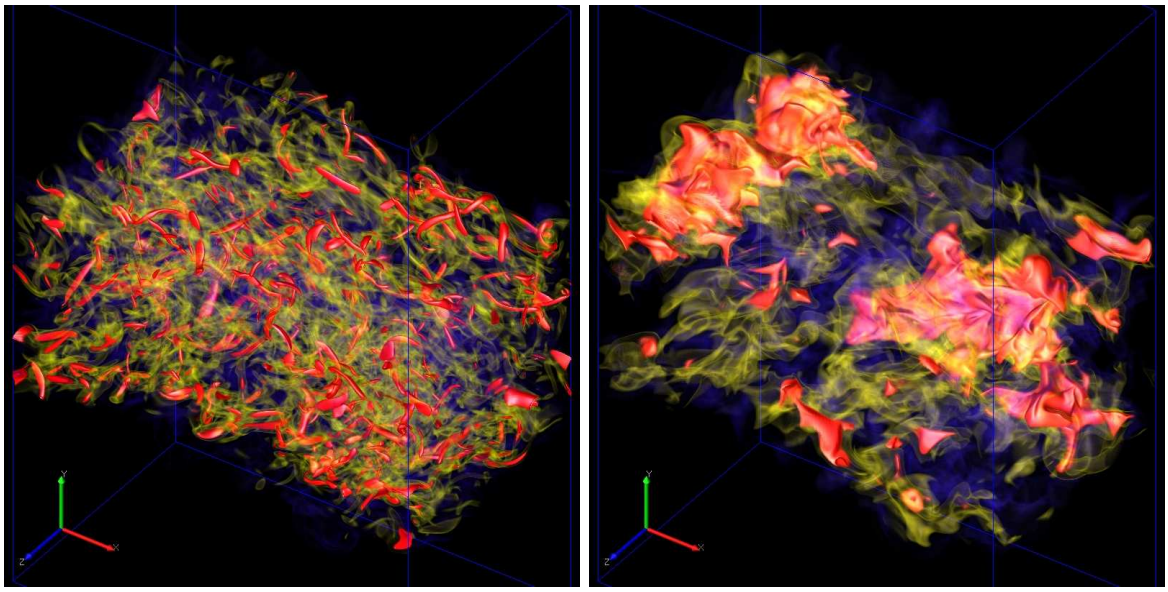

Figure 13: (a) Volume rendering of vorticity magnitude (red, yellow to blue is decreasing intensity) (b) Volume rendering of OH dissipation rate (red, yellow to blue is decreasing intensity)

flow are governed by the range of turbulence scales as well as the range of flame or chemical scales and hence complex. The computational requirements of a nonreacting isotropic turbulence are much more easier to evaluate and hence is used as the starting point to predict the requirements of a reacting flow DNS.

Consider the cost of a DNS of nonreacting isotropic turbulence in a box. The domain size must be large enough to represent the energy-containing motions; and the grid spacing must be small enough to resolve the dissipative scales. In addition, the time step used to advance the solution must be sufficiently small to accurately resolve the fluid motion. For isotropic turbulence with a given spectrum, a reasonable lower limit on the domain size is eight to ten integral length scales (L_t). In directions of the domain where the flame and fluid flow are statistically homogeneous, typically periodic boundary conditions are imposed. As a consequence of this artificial boundary condition, if the box or domain size is too small, the autocorrelation function of the velocity field may not vanish as it would if the domain size was infinite in extent. The resolution of the smallest, dissipative motions, characterized by the Kolmogorov scale, η, requires a sufficiently small grid spacing such that at least half a grid point point is used to resolve the Kolmogorov scale. The two spatial resolution requirements mentioned above determine the total number of grid points required as a function of the Reynolds number (measure of the ratio of inertial scales to the viscous scales in a fluid flow). For isotropic turbulence, $L_t/\eta \propto Re^{3/4}$. Therefore, the

total number of grid points increases as $N^3 \propto (L_t/\eta)^3 \propto Re_L^{9/4}$ where N is the number of grid points in each of the three directions and Re_L is the Reynolds number based on the turbulence integral length scale.

For the time advancement to be accurate and stable, the CFL condition for the explicit time integrator requires that the time step be small enough such that the pressure waves move only a fraction of the grid spacing per time step. Therefore, $\Delta t \propto \Delta x/C$, where C is the speed of sound. The total duration of the simulation scales with the flow-through time given by L/U, where L is the length of the domain in the main flow direction and U is a characteristic flow velocity. Then the number of time steps required is proportional to $(L/\Delta x)(C/U)$, or N/Ma, where $Ma = U/C$ is the Mach number. Therefore the total cost of the system scales as

$$\text{Cost} \propto \frac{N^4}{Ma} \propto \frac{Re^3}{Ma}$$

Next, the discussion can be extended to the case of a reacting flow DNS to understand its cost scaling. DNS of turbulent combustion involves additional complexities that are listed below.

If the chemical model has K_{sp} species, then the number of equations to be integrated at each point for a 3D simulation is $(4 + K_{\text{sp}})$, as shown earlier in section 2.7.1. This has to be included in the cost estimate as,

$$\text{Cost} \propto (4 + K_{\text{sp}})\frac{N^4}{Ma} \propto (4 + K_{\text{sp}})\frac{Re^3}{Ma}$$

Higher chemical complexity will therefore lead to higher cost. Although the above relation predicts a linear dependence on the number of species involved, there are other factors as well. The number of chemical reactions in the chemical model will, in the worst case, scale as ${}_{K_{\text{sp}}}C_2$ or $O(K_{\text{sp}}^2)$. Since, evaluation of chemical source terms is one of the major factors in the cost, this will pose an additional overhead. The stiffness in the chemistry might limit the integrator time steps to be much shorter than that dictated by the CFL condition. Let t_c denote the chemical time scale representative of the fastest chemical transients. Then the number of time steps is determined by the ratio of the flow through time to t_c. The cost, in this situation, will not be determined by, but instead scales as:

$$\text{Cost} \propto (4 + K_{\text{sp}})\frac{N^3 L}{U t_c} \propto (4 + K_{\text{sp}})\frac{Re^{9/4} L}{U t_c}$$

In the case of turbulent combustion, both the fine scales of turbulence and the reaction zone have to be well resolved. The grid spacing will therefore depend on the lesser of the two quantities: Kolmogorov scale and flame resolution. For example, in a premixed turbulent combustion situation, let l_δ denote the width of the reaction zone. Then a non-dimensional number, Ka, known as the Karlovitz number, can be defined based on l_δ and the Kolmogorov scale, η, as $Ka = (l_\delta/\eta)^2$. For cases, where the resolution requirement of the reaction zone is more than that required for the turbulence, the cost increases depending on

the Karlovitz number. The number of grid points required is then given as, $N \propto L_t/l_\delta = Re_L^{3/4} Ka^{-1/2}$. Then

$$\text{Cost} \propto (4 + K_{\text{sp}}) \frac{N^4}{Ma} \propto (4 + K_{\text{sp}}) \frac{Re^3}{MaKa^2}$$

As the Karlovitz number gets smaller, the cost increases. Similarly, in the case of non-premixed combustion, the resolution requirement of the reaction zone structure can increase the cost depending on another non-dimensional number called the Damköhler number, which is the ratio of chemical to fluid dynamic scales.

From these cost estimates, it can be seen that the Reynolds number and the resolution requirement of the reaction layer are the main cost-determining factors. Given the current capability, a non-premixed turbulent combustion simulation at a jet Reynolds number of 6000 can be performed within the INCITE allocation of 2.5 million hours on the IBM Seaborg at NERSC. Note that in this case, the problem has been chosen such that the resolution requirement for the chemical reaction zone and the smallest scales of turbulence are identical, so as to maximize the Reynolds number achievable within this cost. It is foreseeable that with a 1000 fold increase in computational capability, it will be possible to simulate turbulent combustion at roughly ten times higher Reynolds number. Or, the increase in Reynolds number can be traded for increase in chemical complexity, such as higher hydrocarbon fuels, or increase in Damköhler number, i.e. thinner flames relative to Kolmogorov turbulence scale, a combustion regime many practical devices operate in. Either of these choices will bring us closer to realistic combustion situations and increase the validity and relevance of the scientific observations. Furthermore, with a 1000 fold increase in computational capability, it will be possible to perform DNS of a lab scale flame in a canonical configuration that mimics the experimental parameters closely. This will help facilitate direct comparisons of experimental and high-fidelity numerical data and help provide breakthrough modeling capability in relevant parameter regimes.

2.18 Code Improvements made in FY05-Q4

2.18.1 Performance improvements on Cray X1E

During Q4, the Phoenix system was upgraded from X1 to X1E. While the X1 MSPs had a peak computation rate of 12.8GF, the X1E MSP have 18GF. Hence the theoretical maximum speed-up due to the upgrade is 1.5. During this period we have continued to improve our code as detailed below and these have resulted in significant savings.

- S3D has a high-order finite difference filter module capable of filtering out high frequency noise generated by high-wavenumber, unresolved flow features. The grid filtering operation is applied at every time step and improves the stability of time integration. The filter module was rewritten in F90 style array syntax for better vectorization.

- As mentioned in Q2 and Q3 reports, most sections of S3D were vectorized except for the transport module, which continued to consume a significant portion of the CPU resources. During Q4 the legacy transport library was rewritten in F90 style array syntax. In the new version, the transport coefficients, such as thermal conductivity, viscosity etc., are evaluated for all points in a given plane simultaneously rather than computing them sequentially. This led to better vectorization and improved performance.

As a result of the improvements made to the transport libraries in Q4 the computational cost decreased to 5×10^{-9} hours/grid point/timestep (on X1E) from 1.5×10^{-8} hours/grid point/timestep (on X1). After accounting for the speedup due to processor upgrade, the net performance improvement during Q4 alone is 50%.

2.18.2 Porting the code to XT3

During Q4, S3D was ported to the CrayXT3 Opteron cluster at ORNL. Since S3D has already been ported to the Opteron cluster at the CRF, Sandia, there were not many significant issues. However, it was unable to call system commands directly from the fortran code. S3D uses system commands to routinely perform data file management, such as creating directories for writing files to and tarring up files. Since fortran style system commands were not feasible, a C-language interface was created for the directory creations. Other system commands were suppressed. After this modification, the code was demonstrated to scale extremely well up to 5120 processors as seen from Figure[14].

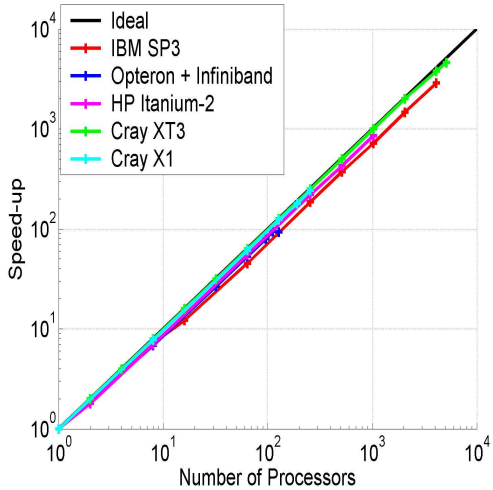

Figure 14: Code scaling on various platforms as of Q4-2005.

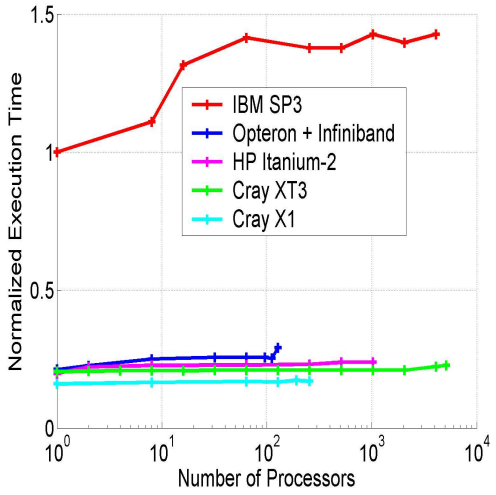

Figure 15: Execution time versus number of processors on various platforms as of Q4-2005.

2.18.3 Summary of Joule target accomplishment

Did your application satisfy the target / constraint? YES!

- During FY2005, S3D's performance was improved on two platforms, the IBM SP at NERSC and the Cray X1E at ORNL. It was also ported to a third platform, the new Cray XT3 at ORNL.

- On the IBM SP, the cost of execution was reduced from 1.5×10^{-7} hours/grid point/timestep at the beginning of FY2005 to 0.8×10^{-7} hours/gridpoint/timestep. The code improvements have improved the efficiency by 45%. Considering that S3D has been used on this platform for several years now, this is a considerable improvement. On the Cray X1E the performance improvement has been truly remarkable. The cost was reduced from 1.8×10^{-7} hours/grid point/timestep to 5×10^{-9} hours/grid point/timestep. After accounting for a factor of 1.5 due to the processor upgrade during this period, the performance improvement is still remarkable.

- The code improvements were found to be beneficial cross-platform. For instance, several modifications made based on profiling the code execution on the IBM SP, where found to be beneficial on the CrayX1E as well. Also, the vectorization of the chemistry module was found to be helpful on the IBM SP as well. The code continues to be portable and efficient on other platforms such as the CrayXT3 at ORNL, the Itanium cluster at PNNL and the local Opteron cluster at Sandia.

- No special compiler options other than the common optimization options were used. The same set of compiler options were used for the entire

source tree without resorting to subroutine-wise compiler tuning.

- The code execution improvement was achieved along with similar improvements to memory usage and parallel performance. As a result the code continues to scale exceptionally well on several Office of Science platforms, as shown in figure[15].

3 AORSA

3.1 Background

Electromagnetic waves play a fundamental role in the dynamics of plasmas [1], both as externally-driven waves and as self-generated instabilities. Plasma waves occur naturally in systems ranging from the solar corona to the planetary magnetosphere and the Earths ionosphere. Plasma waves also occur in laboratory and commercial devices, but it is in the complex phenomena that arise in fusion-relevant laboratory plasmas that understanding of wave dynamics poses the most significant scientific challenge. The plasmas in fusion devices are bounded and inhomogeneous, and have essential variations in at least two spatial dimensions (2-D) and in many cases three dimensions (3-D). In the presence of a confining magnetic field, these plasmas are also anisotropic. Within fusion plasmas, there can be many distinct wave modes propagating at a given frequency in a single region of space. These different wave modes can coexist with widely separated wavelengths, or they can interact with each other when spatial variation of the plasma brings the wavelengths close together. This process is called *mode conversion* [2]. There are numerous mechanisms by which the waves can be absorbed: collisional damping, collisionless Landau and cyclotron damping, and damping by collisionless stochastic interactions. In addition, the waves can produce nonlinear modifications of the plasma medium, thereby altering the wave propagation characteristics and changing the macroscopic properties of the plasma. This in turn can affect other processes occurring in the plasma such as stability and transport, where the modifications can be at the microscopic level of the energy distribution function. It is important to the future advancement of wave-plasma interaction physics that all of these wave and plasma effects be self-consistently analyzed in the appropriate, multi-dimensional geometry.

While complex plasma wave processes are interesting in themselves, understanding these processes can also have practical significance. Since the beginning of the fusion program, it has been recognized that auxiliary heating and current drive will be required to create burning plasma conditions. Tremendous progress has been made both experimentally and theoretically in developing techniques that utilize electromagnetic waves in the radio frequency (rf) range for auxiliary heating and noninductive current drive, and more recently to drive plasma flows [3] that enhance confinement [4, 5]. Construction of the International Tokamak Experimental Reactor (ITER) [6] has been designated as a top priority for new research facilities by the DOE Office of Science. The present plans for ITER call for rf heating in the ITER core plasma, and there are many

additional opportunities to optimize ITER performance with wave control. Improved theory and computational modeling are needed to achieve these goals.

A complete theory for the interaction of rf waves with plasma can be divided into separate but interacting pieces as indicated schematically in [16]. First, the 3D launching structure and coupling of this structure to the plasma must be modeled with sufficient realism and detail to reliably predict the spectrum of the launched waves, the intense fields near the antenna, and the interaction of these fields with the edge plasma. For this step, a 3-D antenna code such as RANT3D [7] can be used.

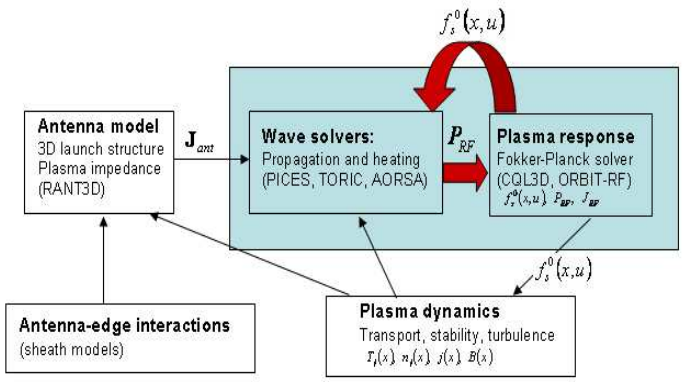

Figure 16: The elements of rf theory been studied independently, or only loosely coupled.

Next, the wave equation must be solved within the plasma to predict the rf electric field structure and the ultimate absorption of the waves by the plasma particles. Examples of wave solvers include TORIC [8], PICES [9], and AORSA [10]. The wave fields induce local macroscopic sources of deposited power, driven current, and plasma flow, all of which are quadratic operators on the electric field [11]. These must be calculated using the full detail of the wave field solution and the necessary structure of the underlying plasma equilibrium.

Finally, the long time response of the plasma distribution function is obtained from the time-averaged form of the Boltzmann equation, or Fokker-Planck equation. Examples of Fokker Planck solvers include CQL3D [12] and ORBIT-RF [13]. All of the elements in Fig. 1 must be integrated at the appropriate level. The conductivity operator is an integral part of the wave solution, whereas the antenna model, the Fokker-Planck solution, and the macroscopic sources can be treated separately and integrated at the level of separate but communicating calculations. Although these elements all interact, they have to a large extent been studied independently. Integration of these elements is a first step toward a comprehensive, integrated wave-plasma simulation capability.

3.2 Physics Basis for AORSA

Time-dependent processes in fusion plasmas are governed by the Maxwell - Boltzmann system of equations [1]. The plasma state is described by a distribution function $f_s(\mathbf{r}, \mathbf{v}, t)$, representing the charged particle density of species s in a 6-D phase space of position $\mathbf{r}$ and velocity $\mathbf{v}$. This function evolves in time according to the Boltzmann equation by convection in the 6-D phase space while under forces exerted by the electric and magnetic fields ($\mathbf{E}$ and $\mathbf{B}$, respectively). The Boltzmann equation is inherently nonlinear because $\mathbf{E}$ and $\mathbf{B}$ arise from charge and electric currents described by velocity moments of $f_s(\mathbf{r}, \mathbf{v}, t)$, as well as from applied external magnetic fields. For rf applications, the wave time scale is by far the fastest time scale in the system. The nonlinear response of the plasma occurs on a much slower collisional, or MHD time scale. Thus, the fields and distribution function can be separated into a time-average or equilibrium part $(\mathbf{E}_0, \mathbf{B}_0, f_s^0)$, and a time-harmonic or rapidly oscillating part, $[\mathbf{E}(\mathbf{r}), \mathbf{B}(\mathbf{r}), f_s^1(\mathbf{r}, \mathbf{v})]e^{-i\omega t}$. For the time-harmonic part, Maxwells equations reduce to a generalization of the Helmholtz equation [1],

$$-\nabla \times \nabla \times \mathbf{E} + \frac{\omega^2}{c^2}(\mathbf{E} + \frac{i}{\omega\epsilon_0}\mathbf{J}_p) = -i\omega\mu_0\mathbf{J}_{ant} \tag{25}$$

plus boundary conditions, where $\mathbf{E}$ is the rapidly oscillating part of the wave electric field and ω is the frequency of the wave. The fluctuating plasma current induced by the wave fields, $\mathbf{J}_p$ is a nonlocal, and possibly nonlinear, integral operator on the electric field [14],

$$\mathbf{J}_p(\mathbf{r}, t) = \int_{-\infty}^{t} dt' \sigma(f_s^0(E), \mathbf{r}, \mathbf{r}', t, t')\mathbf{E}(\mathbf{r}', t') \tag{26}$$

where $\sigma(f_s^0(E), \mathbf{r}, \mathbf{r}', t, t')$ is the *plasma conductivity kernel*. The source for the waves is an externally driven antenna current, $\mathbf{J}_{ant}$, localized near the plasma edge. The rf interaction takes place in a bounded domain determined by the shape of the fusion device. This domain can be very complex, particularly near an antenna structure or other experimental equipment. Computationally, the solution of the wave equation is an extremely intensive task because of the nonlocal nature of the conductivity operator, the geometric complexity of the plasma boundary, and the enormous range of spatial scales that must be treated [2]. These scales range from several meters for toroidal wavelengths of the fast magnetosonic wave, to less than a millimeter for ion Bernstein waves (IBW) or for nonlinear sheaths at the plasma edge.

Recently, great strides have been made by increasing the speed, resolution, and physics content of our wave solvers [15,16]. New wave solvers have been developed in 2- and 3-D called all-orders spectral algorithms (AORSA) [2,10,16]. Previous full-wave models for radio frequency heating in multi-dimensional plasmas have relied on either cold-plasma or finite Larmor radius approximations. These previous models assume that the perpendicular wavelength of the rf field is much larger than the ion Larmor radius, and they are therefore limited to

relatively long wavelengths and low cyclotron harmonics. The AORSA model eliminates these restrictions. By taking advantage of computational techniques for massively parallel computers, AORSA solves the more general integral form of the wave equation with no restriction on wavelength relative to orbit size and no limit on the number of cyclotron harmonics. With this approach, the limit on attainable resolution comes not from the model, but only from the size and speed of the available computer. As a result, it is now possible, for example, to study the mode conversion process in 2-D for large-scale fusion devices [2]. In addition, AORSA has now been extended to include non-thermal plasma species that arise from neutral beam injection, fusion reactions, and wave-driven acceleration of the resonant plasma species.

The goal of our present research is to begin to make the linkage between the new AORSA wave solvers and the Fokker Plank solvers [12, 13] as shown in the blue box in Fig[16]. For most applications, the rapidly oscillating wave fields are small enough compared to the equilibrium fields that the equations can be linearized with respect to the field amplitudes. The rapidly varying part of $\mathbf{E}$ is used in (26) to get the fluctuating wave current. The long time-scale response of the plasma distribution function $f(\mathbf{r}, \mathbf{v}, t)$, is obtained from the Fokker-Planck equation [12, 13],

$$\frac{df}{dt} = \nabla_{\mathbf{u}} \cdot \Gamma_{\mathbf{u}} + R(f) + S \tag{27}$$

where df/dt is the total time derivative following the particle guiding center, and R and S are radial diffusion and particle source/sink operators, respectively. The velocity space flux vector $\Gamma_{\mathbf{u}}$ incorporates two components such that $\nabla_{\mathbf{u}} \cdot \Gamma_{\mathbf{u}} = C(f_0) + Q(\mathbf{E}, f)$ where $C(f_0)$ is the nonlinear operator describing collisions, and $Q(\mathbf{E}, f)$ is the quasilinear operator describing the wave-induced velocity space diffusion of f [17]. The difficulty inherent in coupling the full-wave rf solution to the Fokker-Planck solution lies in evaluating the quasilinear operator $Q(\mathbf{E}, f)$ directly from the full-wave rf electric fields. This has only recently been accomplished, and the formalism is described in 3.2.3.

3.2.1 Plasma conductivity tensor for non-Maxwellian components

The high-performance plasma envisioned for burning plasma devices, such as ITER, will contain significant concentrations of nonthermal plasma species arising from neutral beam injection, fusion reactions, and wave-driven acceleration of resonant plasma species. Initial studies in 1-D [18–20] as well as experimental results, indicate that these nonthermal components can significantly alter the wave propagation and absorption processes. Radio-frequency-driven modifications of the resonant particle velocity distributions have been observed to have a significant impact on the stability and confinement properties of the plasmas in experiments on the Tokamak Fusion Test Reactor (TFTR) [21] and the Joint European Torus (JET) [22] as well as other tokamaks. Also, rf-driven particle fluxes, dominated by energetic, nonthermal particles, can exert torque on the plasma, affecting rotation and stability [4, 5]. Thus, an accurate, self-consistent

calculation of the rf fields consistent with the plasma distribution is a requirement.

To compute self-consistent wave fields and particle distribution functions, four significant physics models must be iteratively coupled:

1. plasma conductivity for non-Maxwellian distribution functions
2. a wave solver incorporating the non-Maxwellian conductivity
3. the QL operator that drives non-thermal distributions
4. a Fokker-Planck solver

AORSA now includes expressions for plasma susceptibility and heating that encompass arbitrary, non-Maxwellian distribution functions.

Following Stix [1], we write the generalized plasma conductivity tensor for an arbitrary non-relativistic species s as

$$\chi_s = \frac{\sigma_s}{-i\omega\epsilon_0} = 2\pi\frac{\omega_p^2}{\omega^2}[\sum_{l=-\omega}^{\omega}\int\frac{du_\parallel}{1-\frac{n_\parallel u_\parallel}{\sqrt{\mu}}-\frac{l\Omega}{\omega}}\int du_\perp U\mathbf{S}_l + \mathbf{e}_\parallel\mathbf{e}_\parallel\int du_\parallel\int du_\perp u_\parallel(u_\perp\frac{\partial f}{\partial u_\parallel}-u_\parallel\frac{\partial f}{\partial u_\perp})] \tag{28}$$

where σ_s is the conductivity tensor, χ_s is the normalized conductivity or susceptibility tensor, and the function U contains derivatives of the distribution function in velocity space,

$$U = \frac{\partial f}{\partial u_\perp} - \frac{n_\parallel}{\sqrt{\mu}}(u_\parallel\frac{\partial f}{\partial u_\perp} - u_\perp\frac{\partial f}{\partial u_\parallel}) = \sin\theta\frac{\partial f}{\partial u} + (\frac{\cos\theta}{u} - \frac{n_\parallel}{\sqrt{\mu}})\frac{\partial f}{\partial\theta} \tag{29}$$

In (28), l is the harmonic number, Ω is the cyclotron frequency, ω_p is the plasma frequency, ω is the rf frequency, and $n_\parallel = k_\parallel c/\omega$ where $k_\parallel$ is the wave number parallel to the magnetic field $\mathbf{B}$. In (29), u and $\theta = \tan^{-1}(u_\perp/u_\parallel)$ are the normalized velocity and the pitch angle, respectively. The tensor $\mathbf{S}_l$ in (28) is defined as,

$$\mathbf{S}_l = \begin{pmatrix} \frac{1}{2}u_\perp^2 J_{l+1}^2(\xi) & \frac{1}{2}u_\perp^2 J_{l+1}(\xi)J_{l-1}(\xi) & \frac{1}{\sqrt{2}}u_\perp u_\parallel J_{l+1}(\xi)J_l(\xi) \\ \frac{1}{2}u_\perp^2 J_{l+1}(\xi)J_{l-1}(\xi) & \frac{1}{2}u_\perp^2 J_{l-1}^2(\xi) & \frac{1}{\sqrt{2}}u_\perp u_\parallel J_{l-1}(\xi)J_l(\xi) \\ \frac{1}{\sqrt{2}}u_\perp u_\parallel J_{l+1}(\xi)J_l(\xi) & \frac{1}{\sqrt{2}}u_\perp u_\parallel J_{l-1}(\xi)J_l(\xi) & u_\parallel^2 J_l^2(\xi) \end{pmatrix} \tag{30}$$

where the velocity u is normalized to $\nu_c = c/\sqrt{\mu}$, c is the speed of light, and $\mu = mc^2/2eE_{norm}$ where E_{norm} is the maximum energy (measured in eV) at which the numerical distribution function is evaluated. The components of velocity perpendicular and parallel to the magnetic field $\mathbf{B}$, are $u_\perp$ and $u_\parallel$, respectively, and the distribution function f is normalized to n/ν_c^3, where n is the density. The argument of the Bessel functions in (30) is $\xi = k_\perp\nu_\perp/\Omega = (k_\perp u_\perp/\Omega)c/\sqrt{\mu}$. (28) has been rotated to a frame in which the wave electric field vector $\mathbf{E}$ is

expressed as $(E_-, E_+, E_\parallel)$ where $E_\parallel$ is parallel to $\mathbf{B}$, and $E_-$and E_+ are the right and left hand circularly polarized field components, respectively,

$$\begin{aligned} E_- &= \frac{1}{\sqrt{2}}(E_\alpha - iE_\beta) \\ E_+ &= \frac{1}{\sqrt{2}}(E_\alpha + iE_\beta) \end{aligned} \tag{31}$$

Here α and β denote the local magnetic coordinates (Stix frame) in which the longitudinal unit vector $\mathbf{e}_\parallel$ points in the direction of the applied magnetic field $\mathbf{B}$, and the two remaining unit vectors, $\mathbf{e}_\alpha$ and $\mathbf{e}_\beta$, are chosen in any convenient direction perpendicular to $\mathbf{B}$. We choose $\mathbf{e}_\alpha$ to be that part of the ξ unit vector $\mathbf{e}_\xi$ that is perpendicular to $\mathbf{B}$. The remaining unit vector $\mathbf{e}_\beta$ is the cross product $\mathbf{e}_b \times \mathbf{e}_\alpha$. The plasma conductivity tensor σ_s is related to the susceptibility χ_s in (28) by $\sigma_s = -i\omega\epsilon_0\chi_s$, and the plasma dielectric tensor is $\epsilon = \mathbf{I} + \sum_s \chi_s$.

The integral over $u_\parallel$ in (28) is singular at cylotron resonance, and is evaluated using the Plemelj relation

$$\int_{-\infty}^{\infty} \frac{F(u_\parallel)du_\parallel}{1 - \frac{n_\parallel u_\parallel}{\sqrt{\mu}} - \frac{l\Omega}{\omega}} = P\int_{-\infty}^{\infty} \frac{F(u_\parallel)du_\parallel}{1 - \frac{n_\parallel u_\parallel}{\sqrt{\mu}} - \frac{l\Omega}{\omega}} - \frac{\sqrt{\mu}}{|n_\parallel|}\pi i F(u_{\parallel,res}) \tag{32}$$

where F is an arbitrary function of $u_\parallel$, P denotes the Cauchy principal parts integral, and $u_{\parallel,res} = (1 - l\Omega/\omega)\sqrt{\mu}/n_\parallel$.

3.2.2 Energy absorption by non-Maxwellian plasmas

In many cases of interest, the energy absorbed by the plasma $\partial W/\partial t$ can be approximated by $1/2Re(\mathbf{E}^* \cdot \mathbf{J}_p)$ where $\mathbf{J}_p$ is the plasma current in (26) calculated from the susceptibility in (28). However, when there is significant wave energy carried by thermal motion of the particles (i.e. kinetic flux), a more general expression for $\partial W/\partial t$ is required. In this case, the local energy absorption can be expressed as [12]

$$\frac{\partial W}{\partial t} = \frac{1}{2}Re\{\frac{\epsilon_0\omega}{i}\sum_{\mathbf{k}_1,\mathbf{k}_2} e^{i(\mathbf{k}_1-\mathbf{k}_2)\cdot\mathbf{r}}\mathbf{E}^*_{k_2} \cdot \mathbf{W}_l \cdot \mathbf{E}_{k_1}\} \tag{33}$$

where $\mathbf{W}_l$ is the local energy absorption kernel,

$$\mathbf{W}_l = 2\pi\frac{\omega_p^2}{\omega^2}\sum_{l=-\infty}^{\infty} e^{il(\beta_1-\beta_2)}\mathbf{C}^{-1}(\beta_2) \cdot \mathbf{\Theta} \cdot \mathbf{C}(\beta_1) \tag{34}$$

and

$$\mathbf{C}(\beta) = \begin{pmatrix} e^{i\beta} & -ie^{i\beta} & 0 \\ e^{-i\beta} & -ie^{-i\beta} & 0 \\ 0 & 0 & \sqrt{2} \end{pmatrix} \tag{35}$$

where β is the angle between $k_\perp$ and k_α such that $cos\beta = k_\alpha/k_\perp$ and $sin\beta = k_\beta/k_\perp$. Note that $\mathbf{C}(\beta)$ is the rotation matrix that transforms the electric field

from Stixs local magnetic coordinates with $k_\beta = 0$ to the $(E_+, E_-, E_\parallel)$ frame. In (34),

$$\mathbf{\Theta}(r, k_1, k_2) = \int_{-\infty}^{\infty} \frac{du_\parallel}{1 - \frac{n_\parallel u_\parallel}{\sqrt{\mu}} - \frac{l\Omega}{\omega}} \int_0^\infty du_\perp U \mathbf{S}' \tag{36}$$

where U is defined in (29), and

$$\mathbf{S}_l = \begin{pmatrix} \frac{1}{2}u_\perp^2 J_{l+1}(\xi_2)J_{l+1}(\xi_1) & \frac{1}{2}u_\perp^2 J_{l+1}(\xi_2)J_{l-1}(\xi_1) & \frac{1}{\sqrt{2}}u_\perp u_\parallel J_{l+1}(\xi_2)J_l(\xi_1) \\ \frac{1}{2}u_\perp^2 J_{l+1}(\xi_2)J_{l-1}(\xi_1) & \frac{1}{2}u_\perp^2 J_{l-1}(\xi_2)J_{l-1}(\xi_1) & \frac{1}{\sqrt{2}}u_\perp u_\parallel J_{l-1}(\xi_2)J_l(\xi_1) \\ \frac{1}{\sqrt{2}}u_\perp u_\parallel J_l(\xi_2)J_{l+1}(\xi_1) & \frac{1}{\sqrt{2}}u_\perp u_\parallel J_l(\xi_2)J_{l-1}(\xi_1) & u_\parallel^2 J_l(\xi_2)J_l(\xi_1) \end{pmatrix}. \tag{37}$$

In writing (33 - 37), we have assumed that only the dissipative part of the resonant integral [second term in (32)] contributes to the power absorption. For this reason, the analog of the $\mathbf{e}_\parallel\mathbf{e}_\parallel$ term in (28) does not appear. Notice that when $k_1 = k_2$, the tensors in (30) and (37) are identical ($\mathbf{S}'_l = \mathbf{S}_l$), and the sums over k_1 and k_2 in (33) can be separated to yield $\partial W/\partial t = 1/2 Re(\mathbf{E}^* \cdot \mathbf{J}_p)$. When $k_1 \neq k_2$, these sums can be extremely time consuming to evaluate. In 2-D for example, four nested do loops are required to evaluate (33), and in 3-D, six nested loops are required. Even for Maxwellian distributions, calculating these sums can take orders of magnitude more computation time than the wave solution itself. For non-Maxwellians, the time is totally prohibitive. Therefore, it is worth some effort to find a more efficient way to calculate the power absorption. Recently we have noticed that by bringing the velocity space integrals in (36) outside of the sum over k_1 and k_2, these sums can be separated, even for $k_1 \neq k_2$ yielding,

$$\begin{aligned} \frac{\partial W}{\partial t} &= \frac{\pi}{2}\frac{\epsilon_0\omega_P^2}{\omega} \times \\ & Im \ \{\int_0^\infty du_\perp \sum_{l=-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{U du_\parallel}{1 - \frac{n_\parallel u_\parallel}{\sqrt{\mu}} - \frac{l\Omega}{\omega}} \left(\sum_{\mathbf{k}_2} \epsilon_{\mathbf{k}_2}^{*T} \cdot \mathbf{a}_l^{(2)T}\right) \cdot \left(\sum_{\mathbf{k}_1} \mathbf{a}_l^{(1)} \cdot \epsilon_{\mathbf{k}_1}\right)\} \end{aligned} \tag{38}$$

where the rotated electric field has been denoted as $\epsilon_{\mathbf{k}} = \mathbf{C}(\beta) \cdot \mathbf{E}_k e^{i(\mathbf{k}\cdot\mathbf{r} + l\beta)}$ or,

$$\begin{aligned} \epsilon_1 &= \frac{1}{\sqrt{2}}(E_\alpha - iE_\beta)e^{i(\mathbf{k}\cdot\mathbf{r}+l\beta)} \\ \epsilon_2 &= \frac{1}{\sqrt{2}}(E_\alpha + iE_\beta)e^{i(\mathbf{k}\cdot\mathbf{r}+l\beta)} \\ \epsilon_3 &= E_\parallel e^{i(\mathbf{k}\cdot\mathbf{r}+l\beta)} \end{aligned} \tag{39}$$

and,

$$\mathbf{a}_l = \left(u_\perp J_{l+1}(\xi), u_\perp J_{l-1}(\xi), \sqrt{2}u_\parallel J_l(\xi)\right). \tag{40}$$

Assuming that only the dissipative part of the resonant integral contributes to the power absorption, (39) yields,

$$\frac{\partial W}{\partial t} = \frac{\pi}{2}\frac{\epsilon_0\omega_P^2}{\omega} Re\{\int_0^\infty du_\perp \frac{\sqrt{\mu}}{|n_\parallel|}\pi U \sum_{l=-\infty}^{\infty} \left(\sum_{\mathbf{k}_2} \epsilon_{\mathbf{k}_2}^{*T} \cdot \mathbf{a}_l^{(2)T}\right) \cdot \left(\sum_{\mathbf{k}_1} \mathbf{a}_l^{(1)} \cdot \epsilon_{\mathbf{k}_1}\right)_{u_{\parallel,res}} \} \tag{41}$$

This expression requires that the velocity space integral over u? be calculated numerically rather than analytically using plasma dispersion (Z) functions [23]. However, when the distribution function is non-Maxwellian, the Z functions are replaced by numerical integrals anyway. Hence, there is no additional work involved. Also, because the sums over k_1 and k_2 are separated, there is an enormous savings in computation time, and (41) can be evaluated in approximately the time required to calculate the plasma current $\mathbf{J}_p$.

3.2.3 Quasi-linear diffusion coefficients

The CQL3D Fokker-Planck code [12] solves the bounce-averaged form of the Fokker-Planck equation, or

$$\frac{\partial}{\partial t}(\lambda f_0) = \nabla_{\mathbf{u}_0} \cdot \Gamma_{\mathbf{u}_0} + \langle\langle R \rangle\rangle + \langle\langle S \rangle\rangle \tag{42}$$

where f_0 is the bounce averaged distribution function evaluated at the outer equatorial plane, and expressed as a function of one spatial variable ρ labeling a flux surface, and two velocity space variables $u_{\perp,0}, u_{\parallel,0}$, referenced to the midplane. In (42), $\langle\langle R \rangle\rangle$ is the bounce-averaged radial diffusion operator, and $\langle\langle S \rangle\rangle$ is the bounce averaged particle source/sink operator. The coefficient λ is defined as $\lambda = |u_{\parallel,0}|\tau_b$, where τ_b is the bounce time. The velocity space divergence of the quasilinear energy flux in 42 can be written as [12]

$$\nabla_{\mathbf{u}_0} \cdot \Gamma_{\mathbf{u}_0} = \frac{1}{u_0^2}\frac{\partial}{\partial u_0}\left(B_0 \frac{\partial f_0}{\partial u_0} + C_0 \frac{\partial f_0}{\partial \theta_0}\right) + \frac{1}{u_0^2 \sin\theta_0}\frac{\partial}{\partial \theta_0}\left(E_0 \frac{\partial f_0}{\partial u_0} + F_0 \frac{\partial f_0}{\partial \theta_0}\right) \tag{43}$$

where B_0, C_0, E_0, and F_0 are the bounce averaged quasilinear diffusion coefficients referenced to the midplane.

To solve the Fokker-Planck equation self-consistently with the full wave rf solution, the quasilinear diffusion coefficients B_0, C_0, E_0, and F_0 must be derived directly from the full-wave rf electric field solution. These coefficients are closely related to the rf heating rate and can be deduced by comparing (41) with the equivalent expression written in terms of the quasilinear operator. The result

is,

$$B = \frac{\epsilon_0\omega_p^2}{8\omega e E_{norm}\delta u_\parallel} Re\left(\sum_{l=-\infty}^{\infty}\frac{\pi\sqrt{\mu}}{n_\parallel}\left(\sum_{\mathbf{k}_2}\epsilon_{\mathbf{k}_2}^{*T}\cdot\mathbf{a}_l^{(2)T}\right)\cdot\left(\sum_{\mathbf{k}_1}\mathbf{a}_l^{(1)}\cdot\epsilon_{\mathbf{k}_1}\right)_{=u_\parallel,res}\right) \tag{44}$$

$$C = \frac{\epsilon_0\omega_p^2}{8\omega e E_{norm}\delta u_\parallel}\times$$

$$Re\left(\sum_{l=-\infty}^{\infty}\frac{\pi\sqrt{\mu}}{n_\parallel}\frac{1}{u_{\parallel,res}\sin\theta}\left(\frac{l\Omega}{\omega}-\sin^2\theta\right)\left(\sum_{\mathbf{k}_2}\epsilon_{\mathbf{k}_2}^{*T}\cdot\mathbf{a}_l^{(2)T}\right)\cdot\left(\sum_{\mathbf{k}_1}\mathbf{a}_l^{(1)}\cdot\epsilon_{\mathbf{k}_1}\right)_{=u_\parallel,res}\right)$$

$$E = \frac{\epsilon_0\omega_p^2}{8\omega e E_{norm}\delta u_\parallel}\times$$

$$Re\left(\sum_{l=-\infty}^{\infty}\frac{\pi\sqrt{\mu}}{n_\parallel}\frac{1}{u_{\parallel,res}}\left(\frac{l\Omega}{\omega}-\sin^2\theta\right)\left(\sum_{\mathbf{k}_2}\epsilon_{\mathbf{k}_2}^{*T}\cdot\mathbf{a}_l^{(2)T}\right)\cdot\left(\sum_{\mathbf{k}_1}\mathbf{a}_l^{(1)}\cdot\epsilon_{\mathbf{k}_1}\right)_{=u_\parallel,res}\right)$$

$$F = \frac{\epsilon_0\omega_p^2}{8\omega e E_{norm}\delta u_\parallel}\times$$

$$Re\left(\sum_{l=-\infty}^{\infty}\frac{\pi\sqrt{\mu}}{n_\parallel}\frac{1}{u_{\parallel,res}^2\sin\theta}\left(\frac{l\Omega}{\omega}-\sin^2\theta\right)^2\left(\sum_{\mathbf{k}_2}\epsilon_{\mathbf{k}_2}^{*T}\cdot\mathbf{a}_l^{(2)T}\right)\cdot\left(\sum_{\mathbf{k}_1}\mathbf{a}_l^{(1)}\cdot\epsilon_{\mathbf{k}_1}\right)_{=u_\parallel,res}\right)$$

Bounce averaging, and transforming the velocity space derivatives from u, θ to u_0, θ_0 at the midplane gives [12],

$$\begin{aligned} B_0 &= \lambda\langle\langle B\rangle\rangle_{bounce} \\ C_0 &= \lambda\langle\langle\frac{\cos\theta}{\psi^{\frac{1}{2}}\cos\theta_0}C\rangle\rangle_{bounce} \\ E_0 &= \lambda\langle\langle\frac{\cos\theta}{\psi\cos\theta_0}E\rangle\rangle_{bounce} \\ C_0 &= \lambda\langle\langle\frac{\cos^2\theta}{\psi^{\frac{3}{2}}\cos\theta_0}C\rangle\rangle_{bounce} \end{aligned} \tag{45}$$

where $\psi = |\mathbf{B}|/|\mathbf{B}_0|$ and $\mathbf{B}_0$ is the magnetic field at the outer equatorial plane where $\theta_{pol} = 0$.

The bounce averages in (45) can be conveniently expressed in terms of flux surface averages. For example, B_0 can be reformulated as

$$B_0 = \lambda\langle\langle B\rangle\rangle_{bounce} = \oint\frac{dl_B}{\psi}\langle|\frac{u_{\parallel,0}\psi}{u_\parallel}|B\rangle_{flux_surface} \tag{46}$$

and similarly for the other bounce-averaged coefficients. Likewise, the flux surface averaged heating rate can be written in terms of the bounce-averaged distribution function f_0 and the midplane velocity space variables,

$$\langle\frac{\partial W}{\partial t}\rangle_{flux_surface} = \frac{-4\pi e E_{norm}}{\oint dl_B/\psi}\int_0^\infty u_{\perp,0}du_{\perp,0}\int_{-\infty}^{\infty}du_{\parallel,0}\left(\frac{B_0}{u_0}\frac{\partial f_0}{\partial u_0}+\frac{C_0}{u_0}\frac{\partial f_0}{\partial\theta_0}\right) \tag{47}$$

This can be compared to the flux surface average of the heating rate in (41) to check the consistency of the bounce-averaged diffusion coefficients calculated by AORSA.

3.3 Numerical Basis for AORSA

3.3.1 Code Description

AORSA is a Fortran-based parallel processing code developed at Oak Ridge National Laboratory. Code development has a long history that reflects the changes in Fortran definitions and conventions. As the Fortran standard and conventions evolved, the new features were often incorporated into the code base. Thus fixed source, Fortran 77 style code shares space with newer Fortran 90 constructs. In particular, some routines are defined within MODULES, ALLOCATE is used to dynamically manage memory, KIND adds flexibility for data typing, etc. A modern build system manages ports to several high performance computing platforms.

The code base consists of approximately 28,000 lines of executable instructions, 5,500 data declarations, and 8,000 comment lines, contained in around 45 files. The compiled executable ranges in size from 1.3 Mbytes on cheetah (AIX, power4, xlf compilers), to 6.1 Mbytes on ram (Linux, IA64, Intel compilers), to 15.3 Mbytes on phoenix (UNICOS, ssp mode, Cray programming environment). This range is a function of, among other things, the optimization strategies and resources employed by the compiler, operating system requirements, and the use of dynamically loaded shared object libraries.

AORSA makes use of two externally developed software libraries, ScaLAPACK and FFTPACK. The ScaLAPACK library is included as a highly optimized component of most high performance computing environments; FFTPACK is typically not, and thus the source must be available.

Parallelism is centered around the solution of a generated double precision complex-valued system of linear equations. This system is solved using the publicly available ScaLAPACK library; in particular routines pzgetrf factors the matrix into upper and lower matrices, which pzgetrs then uses to compute the solution vector. Inter-process communication within ScaLAPACK is managed using the BLACS component library, which also provides a convenient mode for distributing the matrix into the required two-dimensional block cyclic format. The BLACS is then used for redistributing the solution vector within AORSA code, as well as setting process barriers. ScaLAPACK also contains the parallel BLAS library, which includes two routines directly called by AORSA: the vector scaling pzscal and vector norm pdznrm2.

The Fast Fourier Transformation is accomplished using the publicly available FFTPACK, developed at NCAR. Approximately one third of the 2,500 lines of Fortran source are linked into AORSA.

Its worth noting that although current computing resources of interest are utilized via the MPI parallel programming model, MPI functionality is not directly exposed. Thus ports to other present and future architectures that

provide other inter-process communication protocols is simply dependent upon the provision of a BLACS library, or with only slightly more work, a linear equation solver.

In addition to the small problem definition input file aorsa2d.in, AORSA reads in three data files: ZTABLE.TXT, cql3d.out, and eqdsk.out. For the 128x128 problem, the file sizes are listed in the following table:

File name	Description	Size (MBytes)
cql3d.out	magnetics field geometry	29.3
eqdsk.out	CQL3D output	1.6
ZTABLE.TXT		10.6

Table 5: AORSA Start-Up Data File Sizes

Output is also fairly small, all written by the root process.

File name	Description	Size (MBytes)
movie[ealpha,eb,wdot]	movie generator data	.39-.79
out15	stdout	.13
out38		14
out_cql3d.coef	CQL3D input	39

Table 6: AORSA Output Sizes

3.3.2 Numerical solution of the wave equation

The original version of AORSA uses a fully spectral technique to solve the integral equation represented by (25) and (26). The rf electric field $\mathbf{E}$ is expanded in Fourier harmonics as,

$$\mathbf{E}(x,y) = \sum_{n,m} \mathbf{E}_{n,m} e^{i(k_n x + k_m y)} \tag{48}$$

where x and y are spatial coordinates, and k_n and k_m are the corresponding Fourier wave numbers. For a locally homogeneous plasma, the convolution theorem is used to recast (26) in terms of Fourier harmonics as,

$$\mathbf{J}_p(x,y) = \sum_{n,m} \sigma(x,y,k_n,k_m) \cdot \mathbf{E}_{n,m} e^{i(k_n x + k_m y)} \tag{49}$$

where $\sigma(x,y,k_n,k_m)$ is the Fourier coefficient of the conductivity kernel, or the plasma conductivity tensor. Using these Fourier representations for $\mathbf{E}(x,y)$ and $\mathbf{J}_p(x,y)$, we can write the wave equation (25) at each point on a spatial mesh (x_i, y_j). The resulting set of linear equations can be solved for the Fourier coefficients of the electric field. This is called the method of collocation and

it avoids convolutions associated with the plasma current, while at the same time, including cyclotron harmonics of arbitrary order. Furthermore, boundary conditions are easy to implement at specified positions in space.

Because all modes in the spectral representation are coupled, the complete solution of (25) and (26) requires calculating and inverting a very large, dense matrix. For example, with 200×200 Fourier modes in 2-D, or equivalently, $50 \times 50 \times 16$ Fourier modes in 3-D, it is necessary to solve 120,000 coupled complex equations. The storage required for the resulting matrix is 230 Gbytes. Such large amounts of memory are seldom available on a single processor. However, with large parallel computers, it is possible to distribute the required memory over many processors, and the distributed matrix can be inverted using ScaLAPACK, a library of high-performance linear algebra routines for distributed-memory, parallel computers. Problem size scales as the square root of the number of processors. Using 576 processors on the Oak Ridge National Laboratory IBM RS/6000 SP computer, up to 120,000 coupled complex equations have been solved, while achieving 0.66 teraflops (660 billion operations per second) during matrix factorization. With a still larger computer, such as the NERSC-Seaborg, significantly more equations can be solved. Computing speed for the matrix factorization scales linearly with the number of processors, and there is no indication of saturation caused by communication between processors.

A question arises concerning the scaling of the required computer size for very large problems. Scaling studies show that the possible problem size increases as the square root of the number of processors on a particular computer. In other words, as the computer size increases, the relative increase in problem size gets smaller. For example, with the original spectral version of AORSA described above, a solution on a 512×512 grid would require 800,000 complex equations, 10 TB of memory, and 5120 processors (assuming 2GB / processor). This is equivalent to the entire Japanese "Earth Simulatior"! The spectral approach has a particular disadvantage in this regard. With a conducting shell boundary, the rf electric field in the shell and in the region outside the shell is zero. The spectral method, however, represents these zeros as the sum of many terms in a Fourier series, and these terms must be calculated, even though the solution is known beforehand. To address this problem, the present version of AORSA transforms the linear system of equations (using fast Fourier transforms) from Fourier space back to configuration space before solving the matrix. This transformation allows the solution matrix to be condensed significantly by eliminating boundary points where the solution is known to be zero. The resulting matrix is more diagonally dominant and much smaller than in Fourier space, and the savings can be quite dramatic. The efficiency gained from this new formulation has allowed higher resolution calculations than previously thought possible.

3.3.3 Loading the solution matrix and calculating the non-Maxwellian plasma conductivity tensor

The solution matrix in AORSA is constructed by calculating the plasma current for each particle species using the conductivity tensor in (28) and then summing over species. This conductivity tensor consists of 2-D velocity space integrals over perpendicular and parallel velocities, $u_\perp$ and $u_\parallel$, respectively. For each particle species and cyclotron harmonic, these integrals must be calculated for each Fourier mode in the wave spectrum at every point in space. Thus a 128×128 mesh size with 4 particle species and 20 cyclotron harmonics requires about 2×10^{10} 2-D velocity space integrals. The matrix load is therefore a very significant fraction of the total computation time. The integral over $u_\perp$ is straight forward and is done using an inline trapezoidal rule. The integral over $u_\parallel$ is singular at cyclotron resonance, and therefore is calculated differently depending on whether there is a resonance present in the domain or not. When there is a resonance present, the $u_\parallel$ integral is evaluated using the Plemelj relation in (32), where the Cauchy principal value of $f(x)/(x-c)$ over a finite interval is calculated using a NETLIB routine contained in subroutine cauchy_ppart2. When there is not a resonance in the domain, the $u_\parallel$ integral is evaluated using an inline Simpsons rule.

3.3.4 Re-calulating the plasma current for each species

After the matrix is solved for the rf electric fields, the plasma current is recalculated for each species. This involves exactly the same calculation as described above for loading the matrix, but because of the large amount of memory required to store the required 2×10^{10} velocity space integrals, AORSA actually recalculates these integrals as a post-processing step. An alternate possibility would be to write the required integrals to disk while loading the matrix, and then reading them back after the matrix is solved.

3.3.5 Calculating the non-Maxwellian power absorption and quasilinear operator

The non-Maxwellian power absorption and quasilinear operator are calculated from (41) and (44), respectively. As with the plasma current, the integral over $u_\parallel$ is singular at cyclotron resonance, and can be evaluated using the Plemelj relation in (32). However, here we assume that only the dissipative part of the resonant integral, or the second term in (32), contributes to the power absorption. Hence the parallel velocity integral can be done completely analytically, and no quadrature is required. The remaining integral over $u_\perp$ is calculated using an inline trapezoidal rule. In order to save computation time and encourage vectorization, the Bessel functions in (40) are pre-computed and stored in a table with 51 entries. When evaluating the integrand in the inner-most loop, the Bessel functions are evaluated by interpolation using this table. This reduces the time required to calculate the non-Maxwellian power absorption and the

quasilinear operator to approximately the time required to calculate the plasma current.

3.3.6 Validating the numerical calculations

The AORSA calculations can be validated in a number of ways. First, energy conservation can be used to determine whether the calculation is converged or not; that is whether there are enough Fourier modes present to resolve the physical processes at work in the plasma. To check energy conservation, the total power absorbed by the plasma can be compared with the real power emitted by the antenna. These quantities are given at the end of the out15 output file, and should agree to two or three significant figures for a converged run. Also, the total power absorbed as calculated from $1/2Re(\mathbf{E}^* \cdot \mathbf{J}_p)$ should agree approximately with that calculated from $\partial W/\partial t$ given by (41). Finally, the radial profile of the power absorption calculated directly from the rf electric fields in (41), should agree with that calculated from the quasilinear diffusion coefficients B_0, and C_0, as given by (47). This is a check on the consistency of the quasilinear operator calculated by AORSA. These profiles are printed at the end of out15 and are also included in out38 for plotting.

A more direct check on the convergence of the Fourier series in AORSA is to look that the spectrum of the three components of the rf electric field. These should show strong peaking at low mode numbers and a gradual or rapid decay at large mode numbers. If the spectrum is still increasing at the larger mode numbers, the calculation is not converged, and more modes are required. In this case, there is usually poor energy conservation as well. Arrays containing the spectrum of modes for all three electric field components are included in the out38 output file for plotting.

Finally, the flux surface averaged radial profile for power absorption can be compared directly with experiment when this measurement is available. Other experimental quantities that can be compared with AORSA include the fraction of power absorbed by each particle species and the total power absorbed by the plasma.

3.3.7 Choosing the hardware allocation for AORSA

The AORSA hardware selection is based primarily on local memory requirements. The most significant memory requirement for AORSA is the storage for the distributed matrix, *p_amat*. Before beginning the matrix factorization, AORSA prints the size of this array in Mbytes. The user must be sure that the combined size of the executable (xaorsa2d) and the *p_amat* array together do not exceed the memory available on each processor. If the memory per processor is exceeded, more processors must be. The number of processors requested is specified in the aorsa2d.in data file by the product of the number of processor rows (nrow) and the number of processor columns (ncol).

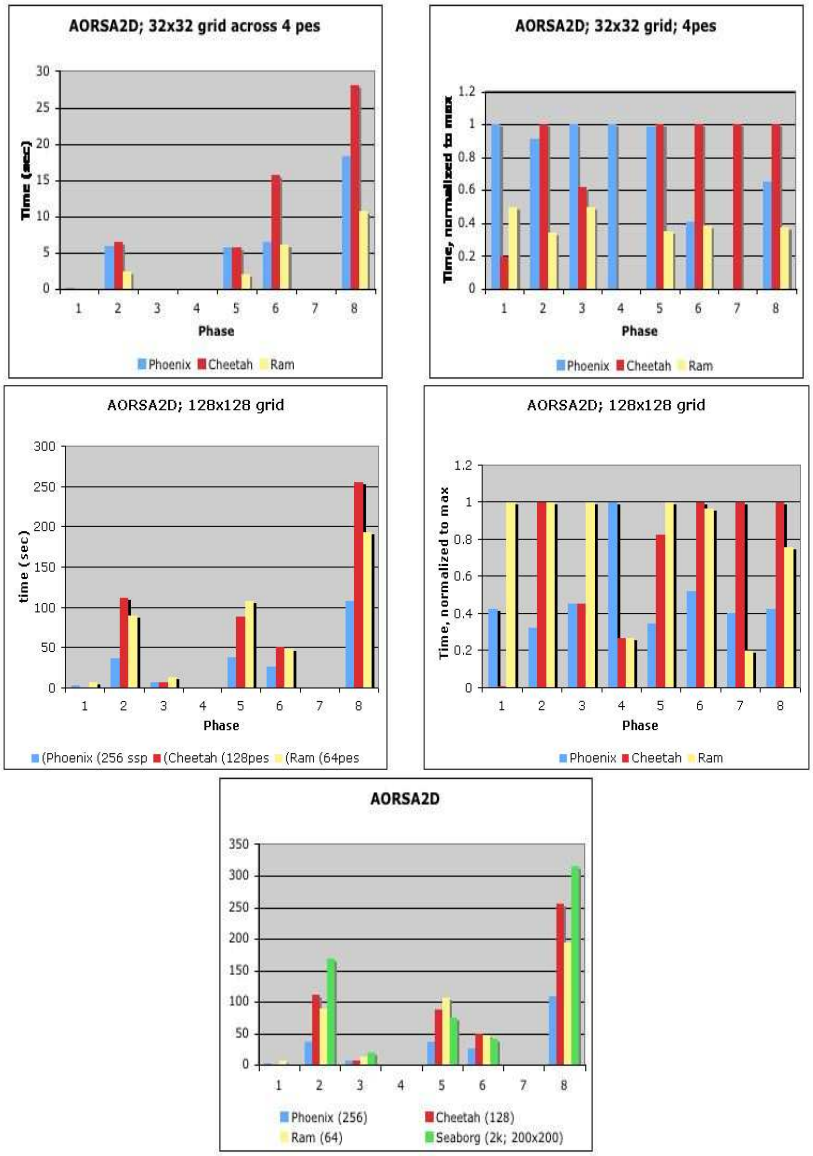

Figure 17: Timings for different sized grids as a function of phases -see Table[7]. There are two plots for the different grid sizes measured: one is the actual time to compute the phase and the other this time normalized to the fastest system. In the bottom plot, the y-axis is not labelled but should be time(seconds). That plot is a replica of the 128×128 plot except that a 2000pe run is included where a 200×200 grid is used.

phase 1	calc U
phase 2	load matrix
phase 3	ScaLAPACK solve
phase 4	FFT
phase 5	calc current
phase 6	compute quasi-lin-op
phase 7	check quasi-lin-op
phase 8	total (1-8)

Table 7: Major phases of aorsa2d for referencing Figure[17]

3.4 Science Case: Non-Maxwellian bulk ion component in the NSTX tokamak

An important problem for rf heating of fusion plasmas is the absorption of rf power by non-Maxwellian components such as minority ion species, fusion-born alpha particles, and fast ions associated with neutral beam injection (NBI). For example, fast ion tails have recently been observed on both the DIII-D [24] and NSTX [25] tokamaks when NBI and ion cyclotron heating are applied simultaneously. In addition, heating of these non-Maxwellian components often occurs at high harmonics of the ion cyclotron frequency where conventional 2-D full wave models for rf heating are not valid. The calculations to be performed in this part of the study are based on ion distribution functions calculated with the CQL3D Fokker-Planck code for a neutral beam injection plasma with high harmonic fast-wave heating in the NSTX tokamak (shot 108251) [25].

3.4.1 Case Study

In this case study, the non-Maxwellian nature of the velocity distribution function has a significant effect on wave propagation as well as on the power absorption. To calculate the plasma current in this case, the standard plasma dispersion function (Z) is replaced by the 2D velocity space integrals in (28), calculated for every mode in the wave spectrum at each point in space. These integrals have been programmed in collaboration with our colleagues at PPPL [18–20], and the required computation time has been reduced by vectorization, parallelization, and in-lining of the required integrals.

3.4.2 Parameters

The input parameters to be specified for this case include:

1. The magnetic field geometry for NSTX shot 108251 is calculated by EFIT and stored in the eqdsk file g108251.00235. This data is processed by the $eqdsk_setup$ code and stored in the file eqdsk.out which is read by AORSA.

2. The non-Maxwellian distribution function for the majority ion species is calculated numerically by CQL3D and stored in the netcdf file $phillips_n stx3.5.2.nc$. This data is processed by the $cql3d_s etup$ code and stored in the file cql3d.out which is read by AORSA.

3. The density, temperature, charge, and mass for all plasma species are specified in aorsa2d.in

4. The rf frequency, power, toroidal mode number, antenna geometry, and number of cyclotron harmonics included are specified in aorsa2d.in.

5. The grid dimensions and the number of Fourier modes used in each dimension are specified in aorsa2d.in.

3.4.3 Observables

The science based observables that the code calculates include:

1. Two dimensional contours of the power absorption ($Watts/m^3$) and the rf wave electric field magnitude (V/m) in configuration space.

2. Ion cyclotron harmonics at which the heating occurs

3. Flux surface averaged heating profiles for each plasma species ($Watts/m^3$)

4. Driven current density profile ($Amps/m^2$)

5. Total power absorbed by each plasma species ($Watts$)

6. Fraction of power absorbed by each plasma species (%)

7. Quasilinear diffusion coefficients $(B_0, C_0, E_0, F_0)(m/s)$. These are changed to cgs units with a dimensional velocity to be used in iteration with CQL3D.

3.5 Science Case: Fusion born alpha particles in the ITER burning plasma

Another important example of a non-Maxwellian plasma component is the fusion-born alpha population in the proposed ITER international burning plasma experiment. The power absorbed by these alpha particles can seriously degrade both the heating and current drive provided by the rf power.

3.5.1 Case Study

In this calculation, the power absorbed by the alpha particles is calculated from the complete non-Maxwellian slowing down distribution function which is provided analytically. Both the electrons and tritium ions are assumed to be Maxwellian. For the assumed frequency of 56 MHz, the power absorbed by the alphas is localized near the Doppler broadened first and second harmonic

resonances which are located just outside the plasma on both the high and low field sides. The alphas account for about 5% of the absorbed power, and the non-Maxwellian nature of their velocity distribution function effects the wave propagation as well as power absorption.

3.5.2 Parameters

The input parameters to be specified for this case include:

1. The magnetic field geometry for the ITER Snowmass case, calculated by EFIT and stored in the geqdsk_iterAT_051002 file. This data is processed by the eqdsk_setup program and stored in eqdsk.out which is read by AORSA.

2. The non-Maxwellian slowing down distribution function for the alpha particles, calculated analytically in the cql3d_setup code, and stored in cql3d.out which is read by AORSA.

3. The density, temperature, charge, and mass for all plasma species, specified in the aorsa2d.in file.

4. The rf frequency, power, toroidal mode number, antenna geometry, and number of cyclotron harmonics included, specified in aorsa2d.in.

5. The grid dimensions and number of Fourier modes used in each dimension, specified in the aorsa2d.in file.

3.5.3 Observables

The science based observables that the code calculates include:

1. Two dimensional contours of the power absorption ($Watts/m^3$) and wave electric field magnitude (V/m) in configuration space.

2. Ion cyclotron harmonics at which the heating occurs

3. Flux surface averaged heating profiles for each plasma species ($Watts/m^3$)

4. Driven current density profile ($Amps/m^2$)

5. Total power absorbed by each plasma species ($Watts$)

6. Fraction of power absorbed by each plasma species (%)

3.6 Q2 Recent Numerical Progress

As mentioned above, the energy absorption is extremely expensive to evaluate using the nested sums in (33). Even for Maxwellian distributions, these sums can take an order of magnitude more time than the wave solution itself. For non-Maxwellians in 2-D, we cant even afford to do the calculation. By bringing the velocity space integrals outside of the summation over k_1 and k_2, we have been able to express the energy absorption as a product of sums rather than as nested sums. The result is an enormous savings in computation time, especially for non-Maxwellians. Below we give some timing comparisons using both the old and new methods.

3.6.1 Energy absorption in 1-D for non-Maxwellian plasmas

We originally tested the new method in one dimension where we could afford to calculate the energy absorption for non-Maxwellians using the nested sums. In this case, the wave solution and plasma current calculation together take about 3 mins on a single IBM processor. The energy absorption calculation, using the old method of nested sums, takes 43 mins giving a total computation time of 46 mins. With the new method, using the product of sums, the energy absorption calculation takes only 3 mins giving a total computation time of 6 mins. Thus for 1-D, we have a factor of 14 improvement in the time for the energy absorption calculation alone, and a factor of 7 improvement in the total calculation time. In 2-D the savings are even greater.

3.6.2 Energy absorption in 2-D for Maxwellian plasmas

Next, we tested the new formulation in two dimensions using 128 processors on the Cray X1. Although we could not afford to do this comparison with non-Maxwellians, we could do it for Maxwellians. In this case, the wave solution and plasma current calculation take about 31 mins, and the energy absorption using the old method takes 235 mins giving a total computation time of 266 mins. With the new method, the energy absorption takes 61 mins giving a total computation time of 92 mins. Thus for Maxwellians in 2-D, we have a factor of 3.8 improvement in the time for the energy absorption calculation and a factor of 2.9 improvement in the total calculation time.

3.6.3 Energy absorption in 2-D for non-Maxwellian plasmas

For non-Maxwellians in two dimensions, the wave solution and plasma current calculation together take about 103 mins on the Cray X1 with 128 processors. The new energy absorption calculation takes 70 mins giving a total computation time of 173 mins. For the non-Maxwellian case, we cannot afford to calculate the energy absorption using the old method. However, we do know that the time to calculate the plasma current is about 45 mins. Thus, the new energy absorption calculation takes only about a factor of 1.6 times longer than the time to calculate the plasma current for a non-Maxwellian in 2-D.

3.7 Q3 Progress and Dynamics

3.7.1 Physics Progress

During the past quarter, we have achieved the first self-consistent solutions for the wave electric field and particle distribution function for ion cyclotron heating in non-Maxwellian plasmas. The focus has been on integrating the AORSA global wave model and the CQL3D Fokker-Planck code. Expressions have been developed for the plasma conductivity and energy absorption in non-thermal, (i.e. non-Maxwellian) plasmas. Quasilinear diffusion coefficients have been derived from the full-wave RF electric fields for use in the Fokker-Planck solver. This new integrated model has been applied to plasmas in the NSTX and ITER tokamaks where absorption of radio frequency (RF) power by non-Maxwellian ions is important. Results for NSTX show the development of an energetic ion tail, even at low RF powers. For powers comparable to experimental values, the ion tail continues to grow without reaching steady state. This suggests that radial diffusion is important and should be included in the Fokker-Planck solution. For ITER, with a 50-50 mixture of deuterium and tritium, an energetic tail develops on the tritium distribution, but the tail is weak and has little effect on the parasitic absorption. For lower tritium densities, more RF power is available per particle, and a more pronounced ion tail develops. At 10the total electron density, the tail accounts for about 12tritium absorption. This work was presented in an invited talk and paper at the 16th Topical Conference on Radio Frequency Power in Plasmas in Park City, Utah, April, 2005 and is in preparation for submission to Nuclear Fusion.

In future work, loss terms due to radial diffusion will be included in the Fokker Planck solution, and the complete time dependence of the energetic tail formation in NSTX will be studied. In addition, the effect of using alternate Fokker-Planck solvers will be explored, for example, the ORBIT-RF code which uses a Monte Carlo method in which orbits are allowed to deviate from a flux surface. The resulting finite orbit effects should lead to energy deposition profiles that are broader than those presently calculated. Finally, results so far have been obtained by passing files back and forth between two different computers. The long term goal is to assemble a stand-alone system (AORSA + CQL3D) in which both codes communicate and interact automatically on the same computing platform.

3.7.2 Numerical Progress

One significant numerical cost in AORSA is the calculation of principal value integrals required for the plasma conductivity. These integrals can be written in terms of logarithms as,

$$(x = x(i)..x(i+1)) \int \frac{dx}{x - x_{res}} = log(\frac{abs(x(i+1) - x_{res})}{x(i) - x_{res}}) \tag{50}$$

Logarithms are especially costly on the IBM, but are quite fast on the Cray.

A modified code has been developed that is more efficient in the calculation of these principal value integrals. This new version includes:

1. Using the logarithm value for two intervals rather than just one. A quadratic interpolation is used in the interval $(x(i-1), x(i), x(i+1))$, so the same *log* value can be used over the entire interval $[x(i-1), x(i+1)]$ rather than only over $[x(i), x(i+1)]$.

2. The log evaluation is dependent only on the mesh point and x_{res}, but not on the function values. Because we commonly compute 6 to 9 integrals with the same mesh and xres, we can pre-compute and reuse the log evaluations.

3. For a uniform mesh, we can convert some of the divisions related to mesh spacing to just multiplies.

4. On the IBM, the vector MASS library (-lmassv -lmass -lessl) for vector evaluation of logs by vlog() is more efficient than the default f90 intrinsic. On the Cray or other machines, vlog.f must be compiled and linked explicitly.

Implementation of the above ideas gives minor differences in the solution because the quadratic interpolated value of log is used on two intervals rather than one. Conceptually, however, the new version should yield accuracy that is similar to the old version. For the medium size NSTX benchmark case (128x128 mesh), the modified code gives a factor of 2.13 speedup on the NERSC Seaborg, a factor of 2.26 speedup on the ORNL Cheetah, a factor of 1.19 speedup on the ORNL Ram, and a factor of 1.17 speedup on the ORNL Cray X1.

Very recently, AORSA has been successfully ported to the ORNL Cray XT-3 (Jaguar). The Joule baseline cases are now being tested on the XT-3. Results so far show that, for the same number of processors (256 in this case), the XT-3 is overall about 3 times faster than the Cray X1 for AORSA. More specifically, ScaLAPACK is two times faster, the plasma conductivity is 3 times faster, and the quasilinear operator is 5 times faster. The changes involved in making AORSA compatible with the XT-3 have made it more portable and uncovered a non-portable coding practice of not explicitly initializing certain complex arrays used in Fourier transforming the electric fields (e.g. fksav).

3.8 Q4 Progress and Dynamics

3.8.1 Physics Progress

During the fourth quarter, we have continued to explore the physics of self-consistent global wave and Fokker-Planck calculations in ion cyclotron heated non-Maxwellian plasmas. We have been invited to present this work at the 2005 Division of Plasma Physics Meeting of the American Physical Society in Denver this October. For this invited talk, we have calculated self-consistent solutions for the wave fields and ion distribution function for fast deuterium ions in the

DIII-D tokamak (shot 122080) at both 60 MHz and 116 MHz. Results show that for 60 MHz, the deuterium ions absorb about 57% of the RF power at the fourth harmonic resonance. About half of this power is absorbed by the energetic tail. At 116 MHz, the deuterium absorbs 95% of the RF power at the eigth harmonic. About 60% of this power can be attributed to the energetic deuterium tail. These results are in approximate agreement with previous ray tracing and global wave calculations that used equivalent Maxwellian fits to the beam ion distribution function, but they are in marked disagreement with experimental data which shows little absorption at the eigth harmonic for 116 MHz. One possible explanation for this discrepancy is that the radial diffusion losses and finite orbit effects, which are neglected in these calculations, play a significant role at the higher frequency.

Figure[18] shows a visualization of the fast deuterium velocity distribution function at f = 116 MHz. Contours in $u_\perp$ and $u_\parallel$ are shown along the minor radius of the tokamak, with the center of the plasma at the lower right side of the image and the edge of the plasma at the upper left. The contours and colors illustrate the changing tail on the ion distribution function as the plasma location changes.

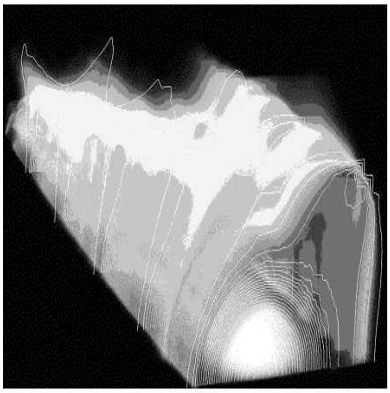

Figure 18: The fast deuterium distribution function in DIII-D shot 122080 at f = 116 MHz. Contours are shown along the minor radius of the tokamak with the center of the plasma at the lower right side of the image and the edge of the plasma at the upper left.

3.8.2 Numerical Progress

AORSA has been enhanced to satisfy the target / constraint for the ORNL `cheetah.ccs.ornl.gov` and NERSC `seaborg.nersc.gov` systems. For the ORNL `ram.ccs.ornl.gov` and `phoenix.ccs.ornl.gov`, we did not satisfy the 50% target, but we did speed up the code by 19% and 17%, respectively. The medium and large test cases no longer run on Phoenix. This is apparently due to the upgrade from the Cray X1 to X1E hardware.

The Joule baseline cases have been tested on the Cray XT-3. Results demonstrate excellent strong scaling as well as very good computational performance on each processor on the XT-3. For a small problem that includes the effect of non-Maxwellian distribution functions (128x128), the runtime is 79.8 minutes on 64 processors, 44.3 minutes on 128 processors, and 24.4 minutes on 256 processors. For a large (256x256) Maxwellian problem designed to study mode conversion in ITER, the runtime is 44.4 minutes on 1024 processors, 27.1 minutes on 2048 processors, and 23.3 minutes on 3072 processors. These times are expected to be meaningfully reduced as Cray focuses on improving the performance of ScaLAPACK on the XT-3. This process will be strongly affected by Crays plans to reduce inter-process communication latencies and collect communication.

For the ITER mode conversion problem, the bulk of computation in AORSA is in the solution of the linear system. For example, the linear solution represents about 68% of the total computation time for a 1024 processor run. The efficiency of this part of the calculation on the XT-3 is about 50% and expected to increase as Cray optimizes ScaLAPACK for the XT-3.

3.9 Publications for the current fiscal year

1. "Integrated Simulation of Fusion Plasmas, D. B. Batchelor, Physics Today (February, 2005).

2. Global Wave Solutions with Self-consistent Velocity Distributions in Ion Cyclotron Heated Plasmas, E. F. Jaeger, R. W. Harvey, L. A. Berry, et al., submitted to Nucl. Fusion (2005).

3. Self-Consistent Full-Wave / Fokker-Planck Calculations for Ion Cyclotron Heating in non-Maxwellian Plasmas, E. F. Jaeger, L. A. Berry, R. W. Harvey, et al., Invited paper, in Proceedings of the 16th Topical Conference on Radio Frequency Power in Plasmas, Park City, Utah, 2005, edited by S. Wukitch (American Institute of Physics, New York, 2005).

4. Self-consistent full-wave and Fokker-Planck calculations for ion cyclotron heating in non-Maxwellian plasmas, E. F. Jaeger, L. A. Berry, R. W. Harvey, et al., Invited paper to be given at the 2005 Division of Plasma Physics Meeting of the American Physical Society, Denver, October (2005), and to be published in Phys. Plasmas.

5. Nonthermal Particle and Full-Wave Diffraction Effects on Heating and Current drive in the ICRF and LHRF Regimes, J. C. Wright, L.A. Berry, P.T. Bonoli, et.al., accepted for publication in Nucl. Fusion (2005).

4 Community Climate System Model -CCSM

The CCSM was described in the last MICS software effectiveness report. That

report is partially (the CCSM section and correlated results) reprinted for reference at the end of this report.

4.1 Overview of the SciDAC CCSM Software Development Project

The DOE Office of Science funded SciDAC project, Collaborative Design and Development of the Community Climate System Model(CCSM) for Terascale Computers, had its initial focus on software engineering for performance portability and model extensibility. This work has been reported in a special issue on climate modeling of the International Journal of High Performance Computing and Applications (IJHPCA, Vol 19, No. 3, August 2005). A special issue of the Journal of Climate (to appear), edited by Bill Collins, reports on various aspects of the CCSM3 models performance in modeling the climate system. Time spent in the early part of the FY05 has been devoted to climate validation and specific optimizations supporting use of the code on the Cray X1 and X1E vector platforms. A broad range of climate change simulations for the US contributions to the Intergovernmental Panel on Climate Change (IPCC) Fourth Assessment Report which will be published in 2007, were performed on the IBM p690(Power4) systems at ORNL and NCAR. The IBM Power3 system at NERSC was also heavily utilized for IPCC runs. In addition, the SciDAC project has worked toward development of new science capabilities in the CCSM modeling system that will be relevant for advanced coupled carbon and climate models which require a chemical atmosphere and dynamic ocean and terrestrial ecosystems. This scientific development work is reported and tracked in the DOE SC GG 5.21.3 JOULE (see http://www.scidac.org/BER/BER_CCSM/asc).

The CCSM is a coupled climate system model consisting of atmosphere, ocean, land and sea ice components. Each model has been developed separately, often with primary developers at different institutions. The coupled system runs within a software framework using a coupler to exchange and regrid outputs and inputs needed for exchange between the components. A typical coupled climate simulation involves integration in time for 100-200 years taking 10 minute timesteps. It is the long integration, as well as the multi-physics complexity of the computation, that qualifies climate simulations as a petascale challenge. Each component is capable of running within the framework with any combination of other components, or what are referred to as data components that couple to observed climatological values. This is important for the advancement of each component model to meet separate validation criteria as well as to be able to function correctly within a coupled system. The behavior of the coupled system is key to identifying feedbacks in the climate system and in providing scientific grounding for decision makers who must address strategies for adaptation to climate change and mitgation of the effects of climate change.

Though the DOE Office of Biological and Environmental Research (OBER) programs are primarily interested in the coupled model for studies of climate change studies on the decadal to century time scales and the prediction of impacts from national energy production choices, the CCSM framework supports

a wide range of scientific research areas including global meteorological teleconnections, interannual and seasonal oscillations, atmospheric radiation, cloud and aerosol feedbacks, ocean and sea ice dynamics, and terrestrial ecosystem processes. The intent of the Community Climate System Modeling project is to support the NSF mission providing a state of the art climate modeling framework to support university research and to support the DOE mission with climate change studies.

4.2 Algorithmic Basis

The Community Climate System Model can be described as a physically based model of the earth's circulations and energy transport. It is based on partial differential equations and physical processes parameterized for the scale of interaction represented in a discrete representation of the atmosphere, ocean, land and sea ice.

Three dynamical cores in the atmosphere model are supported. Each provides a unique capability, but each also has deficiencies. The finite volume dycore is being developed to support simulations that must incorporate chemical cycles and maintain conservation of chemical species. It uses explicit subcycling for fast waves and a long timestep conservative semi-Lagrangian transport scheme. For climate change simulations, the primary atmospheric component is based on an Eulerian spectral dycore which solves the semi-implicit (fast waves) system using fast Fourier transforms and Legendre transformations in a spherical harmonic pseudo-spectral discretization. Two dimensional parallel decompositions of the atmospheric physics calculations are load balanced across the available processors.

The primitive equations for the prognostic variables with a generalized vertical coordinate η in a vorticity-divergence form can be written (with only quadratic nonlinear terms),

$$\frac{\partial \zeta}{\partial t} = \mathbf{k} \cdot \nabla \times \left(\frac{\mathbf{n}}{\cos \phi} \right) + F_{\zeta H}, \tag{51}$$

$$\frac{\partial \delta}{\partial t} = \nabla \cdot \left(\frac{\mathbf{n}}{\cos \phi} \right) - \nabla^2 \left(E + \Phi \right)) + F_{\delta H}, \tag{52}$$

where

$$n_U = (\zeta + f)V - \dot{\eta} \frac{\partial p}{\partial \eta} \frac{\partial U}{\partial p} - R \frac{\Theta}{a} \frac{\pi}{p} \frac{\partial p}{\partial \pi} \frac{\partial \Pi}{\partial \lambda} + F_U, \tag{53}$$

$$n_V = -(\zeta + f)U - \dot{\eta} \frac{\partial p}{\partial \eta} \frac{\partial V}{\partial p} - R \frac{\Theta \cos \phi}{a} \frac{\pi}{p} \frac{\partial p}{\partial \pi} \frac{\partial \Pi}{\partial \phi} + F_V. \tag{54}$$

The pressure coordinate vertical velocity is $\omega = \frac{Dp}{Dt}$, $\hat{\omega} = \dot{\eta} \frac{\partial p}{\partial \eta}$, and the material derivative is

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla + \hat{\omega} \frac{\partial}{\partial p}. \tag{55}$$

In these equations the longitude-latitude (λ, ϕ) coordinates are used in the horizontal and $(U, V) = (u, v)\cos\phi$ with

$$E = \frac{u^2 + v^2}{2\cos\phi}. \tag{56}$$

The surface pressure is represented by $\Pi = \ln(\pi = p_S)$ and the virtual temperature Θ is related to both temperature and moisture. Source terms in the equations represent the "physics" calculations and the sub-grid scale parameterizations.

The energy equation written with temperature as the variable is,

$$\frac{DT}{Dt} - \frac{\kappa T \omega}{p} = \frac{Q}{C_p} \tag{57}$$

$$\frac{\partial}{\partial \eta}\left(\frac{\partial p}{\partial t}\right) + \nabla \cdot \left(\mathbf{v}\frac{\partial p}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = 0, \tag{58}$$

$$\frac{\partial \Phi}{\partial \eta} = -\frac{RT}{p}\frac{\partial p}{\partial \eta} \tag{59}$$

The hydrostatic relation for the geopotential is

$$\Phi = \Phi_S \ + \ R\int_{p(\eta)}^{p(1)} \Theta d \ln p \tag{60}$$

The diagnostic equations can be expressed as

$$\omega \ = \ \frac{\partial p}{\partial \pi}\pi \mathbf{V} \cdot \nabla \Pi - \int_0^{\eta} \pi \mathbf{V} \cdot \nabla \Pi d\left(\frac{\partial p}{\partial \pi}\right) \ - \ \int_{p(1)}^{p(\eta)} \delta dp. \tag{61}$$

Also,

$$\begin{aligned} \hat{\omega} \ &= \ \frac{\partial p}{\partial \pi}\left[\int_0^1 \pi \mathbf{V} \cdot \nabla \Pi d\left(\frac{\partial p}{\partial \pi}\right) \ + \ \int_{p(1)}^{p(\eta)} \delta dp\right] \\ &- \ \int_0^{\eta} \pi \mathbf{V} \cdot \nabla \Pi d\left(\frac{\partial p}{\partial \pi}\right) \ - \ \int_{p(0)}^{p(\eta)} \delta dp. \end{aligned} \tag{62}$$

The prognostic equation for Π is

$$\frac{\partial \Pi}{\partial t} \ = \ -\int_0^1 \mathbf{V} \cdot \nabla \Pi d\left(\frac{\partial p}{\partial \pi}\right) \ - \ \frac{1}{\pi}\int_{p(0)}^{p(1)} \delta dp. \tag{63}$$

These are the form of the flow equations used in the Community Atmospheric Model (CAM) Spectral Eulerian dynamical core. The Finite Volume dynamical formulation is based on a Lagrangian vertical coordinate and a flux form of the horizontal equations to allow a conservative semi-Lagrangian transport. The Finite Volume Dycore has been widely adopted in FY05 for use in simulations

that involve chemical tracers and dynamic ecological processes. This constitutes a major algorithmic shift in the efforts of CCSM developers and optimization efforts will be reflected in the results reported here.

In CCSM3 coupled mode, all components, including the coupler are distributed using a multiple binary launch of executables across the processors of the parallel computer. Both shared memory and distributed memory parallelism are supported in each component, for pure OpenMP, pure MPI or a hybrid execution. Vectorization and cache friendly data structures that are adjustable by the user provide much of the performance portability of the CCSM. The project has been able to support the user community and ongoing simulations with a single source code that runs well on all the supported vector and cache based computer platforms. This single source is kept under strict version control and developer access is carefully monitored to maintain the integrity of the model supporting its broad science application, the functionality on multiple target simulation platforms, and the intellectual property of scientists investing their research efforts to advance climate simulation science. The version control system also allows us to"check out" old versions of the code and document when optimization improvements were made. The version control includes the entire build system on each target platform as well as the required input files for initialization and time dependent boundary and forcing data. Thus, modulo the availability of specific computer systems, all runs are reproducible. This is the before vs after methodology used in this report to document a *larger that 50% improvement in simulation time on a fixed computing platform.*

Efforts in FY05 by the SciDAC Consortium, have shifted from optimization for the IBM systems to use of the Cray X1, X1E and Cray XT3 systems. The shift from the Eulerian dynamical core to the Finite Volume atmospheric dynamics has also taken place in this FY, though all capabilities of the CCSM remain supported on all the target platforms.

4.3 CCSM Science Cases

In past reports, we have discussed the science cases in terms of components. The last report outlined two case studies based on the CAM atmospheric code and the POP ocean code run in standalone mode. (The CAM code usually runs the land model for correct simulated surface conditions even in what is called standalone mode.) In this report, we will discontinue the reporting on the POP ocean code and add a new science case based on the fully coupled mode (B-series experiments) of the CCSM code. It is this mode which corresponds to the additional complexity of modeling the full complement of greenhouse gasses in a coupled simulation mode for the IPCC Fourth Assessment. The CCSM simulates climate using a resolved diurnal cycle in both the ocean and atmosphere. Seasonal variability with land surface vegetation changes (growth of leaves, etc) are included and the model reproduces the dynamical behavior of the coupled ocean- atmosphere system with a realistic El Nino Southern Oscillation (ENSO) and other inter-annual to decadal climate oscillations.

A further change in our case studies is also evident with the addition of

tracking the Finite Volume dynamical core (FVcore) of the CAM3. This new dynamical core represents a major algorithmic shift in atmospheric modeling and we report optimization work on this new algorithm that resulted in a factor of 2.1 improvement in throughput. According to the JOULE report criteria, this exceeds the target performance for time to solution in a full atmospheric model, $\frac{(T_0-T_1)}{T_0} = 0.53$.

4.4 Case Study: Baseline Community Atmospheric Model

The Community Atmospheric Model (CAM3) is the sixth generation of the NCAR atmospheric general circulation model. In contrast to previous generations of the atmospheric model, CAM3 was designed through a collaborative process with users and developers in the Atmospheric Model Working Group (AMWG). The AMWG includes scientists funded by the DOE SciDAC project from NCAR, the university community, and DOE laboratories. For CAM3, a variety of dynamical cores are supported. The spectral Eulerian dynamical core is the traditional choice, especially for climate change simulations. But the CAM3 code includes the option to run with semi-Lagrangian spectral dynamics or with Finite Volume dynamics. Several grid resolutions have also been supported with T85 (1.4 degrees) now standard for the spectral dycores and C-grid (1x1.25 degree) for the FVcore. The FV dynamical core has become much more important in FY05 with the introduction of chemical tracers and the MOZART full tropospheric chemistry package into the modeling system. (This was also partially funded by SciDAC.) The conservative properties of the Finite Volume and flux based semi-Lagrangian algorithm are important in chemical simulations.

The simulation metrics reported for the Eulerian Spectral model at T85 and the Finite Volume (C-grid) do not include the one hundred or so chemical constituents of the full model. Much effort has gone into the FVcore partly because of the projected cost of simulations with the full chemistry. We can report significant speedups in both dycores, but call your attention to the FV results in particular. The "stepon" times reported are used for "time to solution metrics". *The time to solution improved by 53%.* These before/after tests were performed using a version of the code dated October 1, 2004, and a recently tagged version in the repository. They are each executed on the Cray X1E located at ORNL using 128 processors (MSPs). We observe a factor of 2.1 improvement in throughput (simulated years per wall clock day) for the Finite Volume version of CAM3. An improvement factor of 1.24 is measured for the Eulerian spectral model.

4.4.1 Parameters

- Horizontal resolution : T85 spectral truncation (128x256)
- Number of vertical levels: 26
- Number of advected constituents: 3

- Stable timestep: 10 min
- Output interval: 1 month
- CCSM CVS tag: cam3.0.19 vs cam3.2.19

4.4.2 Observables

- Throughput: Simulated years per wall clock day : YPD
- Average time (sec) in dynamics per day: dynpkg
- Average time (sec) in physics/chemistry per day: physics
- Average time (sec) in land model per day: land
- Average time (sec) in dynamics- physics data transpose per day: dp
- Average time (sec) in atm-land communication per day: cl2ck
- Average time (sec) to simulate a day: stepon

Table 8: Cray X1E with 128 MSPs, collected September 2005

CAM3 Version	YPD	dynpkg	physics	land	dp	cl2ck	IO	stepon
cam3.0.19 (EUL-T85)	17.5	3.31	7.76	1.07	0.39	0.43	0.06	13.5
cam3.2.19 (EUL-T85)	21.7	2.38	5.74	1.29	0.34	0.59	0.31	10.9
cam3.0.19 (FV-C grid)	8.8	18.64	5.77	0.34	1.78	0.11	0.07	27.0
cam3.2.11 (FV-C grid)	18.8	6.48	4.57	0.40	0.74	0.10	0.05	12.5

4.5 Case Study: The coupled Community Climate System Model

The IPCC Simulation project used the latest, state-of-the-art development of the CCSM and its component models in what may have been the largest, unclassified simulation project ever attempted. The DOE mission to provide policy makers with a science based projection of variety of possible climate futures based on carbon dioxide emission scenarios that correspond to possible economic and technological futures, was approached with a carefully designed set of CCSM climate simulations. The standard case is termed 'business as usual' and consists of a one percent per year increase in atmospheric CO2 with the consequent warming of the earth's land and ocean surfaces. Other scenarios are based on carbon stabilization at specified levels and explore the resulting climates. In order to prepare the CCSM for use in IPCC Scenario studies a variety of things needed to be added to the code. These included historical volcanic

dust and sulfate emissions, GHG emissions, sulfate chemistry, and cloud water prognostics.

The simulations were carried out using the June 2004 release of the CCSM3.0 code. Participating computing centers included the NCAR Climate System Laboratory (CSL), the ORNL Center for Computational Sciences(CCS), the LBNL National Energy Research Supercomputing Center (NERSC) and the Japanese Frontier Center with the Earth Simulator. The runs were coordinated by NCAR's Lawrence Buja over a one year period ending in January 2005. Over 100 TBytes of model results have been made available to the international community through the LLNL Program for Climate Model Diagnosis and Intercomparison (PCMDI) using grid software developed by SciDAC in the Earth System Grid project. To date over 200 scientific papers have been submitted to peer reviewed journals based on these runs and provided as input to the IPCC Fourth Assessment Report. This entire project was carried off in a collaborative spirit with significant contributions of computer resources and research commitment from DOE, NSF, NASA and the Japanese CRIEPI. The US effort based around the CCSM3 model will be of high value in bounding the expected levels of global warming and informing effective responses to climate change.

The production IPCC runs at the ORNL CCS were performed on the IBM p690 (Power4) computer and the observed throughput of the standard configuration was 3.5 simulated years per day (YPD). Unfortunately, the Cray X1 system was not reliable and stable until well into these runs and has only been used subsequently for follow on studies and new high resolution explorations for downscaling IPCC results. The considerable improvement in simulation throughput when moving to the Cray X1 is similar to the performance of the Japanese Earth Simulator. The runs reported here used 248 MSPs of the Cray X1. Validation of the model results to certify their validity on the new machine requires that model climate statistics match the previous results of the same code. With each new upgrade of the Cray programming environment (PE) the validation must be re-certified to avoid system software regression. Performance may either degrade or improve with these releases. As seen Table 2, the October 2004 programming environment (PE5.1) was validated with best throughput performance of 17.49 simulated years per day on 248 processors (MSPs). With the newer 5.3 programming environment, optimization levels were significantly reduced in order to maintain correct answers. The newest version of the code has recovered optimization levels.

4.5.1 Parameters

- Horizontal resolution : T85 spectral truncation for atm, 1 degree for
- Number of vertical levels: 26 atm, 40 ocn
- Number of advected constituents: 11
- Stable timestep: 10 min

- Processor Configuration: 248 PEs - 128*1 atm, 64*1 lnd, 8 ice, 24 ocn, 24 cpl
- CCSM Repository tag: ccsm3_ornldev_brnchT_validation_testing05
- Operating system version: Unicos

4.5.2 Observables

- Throughput: Simulated years per wall clock day : YPD

Table 9: CCSM3 T85 on Cray X1 with 248 MSPs

old (PE5.1)	newer (PE5.3)	newest (PE5.3)
17.49	16.8	17.26

4.6 Case Study: The Parallel Ocean Program

This case study of the POP 1.4.3 code, has been discontinued. This code is currently being used as a benchmark and no algorithmic changes are being made. Hence, there will be no improvement on a fixed platform. Ocean code development at the LANL Center for Ocean and Sea Ice Modeling (COSIM) has focused on a new hybrid vertical coordinate system and the resulting code is called HYPOP. This code is still under scientific development and not production ready.

5 Omega3P

5.1 Introduction

The International Linear Collider (ILC) [37] is a proposed future international particle accelerator for high energy physics research. It would allow scientists to address many of the most compelling questions of this century about dark matter, dark energy, extra dimensions and the fundamental nature of matter, energy, space and time. The ILC would create high-energy particle collisions between electrons and positrons, their antimatter counterparts. From its inception, this billion-dollar scale facility would be designed, funded, managed and operated as an international scientific project. Currently the ILC accelerator Research & Development is being carried out by physicists and engineers from Asia, Europe and the Americas.

A critical issue in the design of the linac for the ILC concerns the stability of the long train of particle bunches that are accelerated to the collision point. Long bunch trains (2820 bunches per pulse) are required to achieve the luminosity necessary for physics discovery with this machine. The electromagnetic

fields generated by the leading bunches, or so called wakefields, can dilute the emittance of the bunch train and in the worse case, can deflect the beam into the wall of the linac, leading to beam breakup (BBU). One major source of wakefields stems from the accelerating cavity in the form of Higher-Order-Modes (HOMs) that are excited by the transiting beam. Some HOMs can interact strongly with the beam to deteriorate beam quality and result in the BBU instability. Therefore, the complete and accurate characterization of the HOMs becomes equally as important as the primary goal of beam acceleration in the accelerating cavity design for the ILC.

Presently the ILC accelerating cavity is based on the Tesla Test Facility (TTF) [38] design which originated from DESY, Germany in the early 90s. A TTF cavity is shown in Fig.1. The geometry consists of seven interior cells and two end cells connected to beam pipes. One HOM coupler is attached to the input end on the left while another HOM coupler plus a fundamental mode (FM) power coupler are attached to the output end on the right. The FM coupler carries the RF power from an external driver into the cavity to accelerate the beam. The two HOM couplers are situated in a way to optimally reduce the effect of the HOMs. To date, HOMs have been studied in two TTF prototypes through expensive, time-consuming cold-test measurements. Because of the complexity of the HOM couplers and the resolution needed to differentiate between modes, numerical modeling of the entire cavity has not been possible using existing simulation software until now.

Figure 19: A prototype of the TTF 9-cell superconducting accelerating cavity for the ILC linac showing one HOM coupler at the input end (Left) and another HOM coupler together with the fundamental mode coupler at the output end (Right).

5.1.1 Parallel Electromagnetic Codes developed under SciDAC

The DOE's SciDAC project, Advanced Computing for 21st Century Accelerator Science and Technology, has supported a well-integrated, multi-institutional, multi-disciplinary team to focus on the large-scale simulations necessary for the design and optimization of electromagnetic systems essential to accelerator facilities throughout the Office of Science complex. Significant progress [46, 47] has been made in the development of a set of parallel electromagnetic codes based on unstructured grid that is aimed at providing the accuracy, speed and problem size not attainable by available tools. Based mainly on the finite-element formulation, the code suite solves the Maxwell's equations in the time and frequency domain with or without particles and covers a wide range of applications in accelerator design (see schematic in Fig. 2). All these codes have been vigorously benchmarked against results from other solvers as well as measured data on actual accelerator hardware.

This new, comprehensive terascale simulation capability in electromagnetics is further enhanced through SciDAC's SAPP and ISICs collaborations in order to solve the most challenging problems facing major DOE accelerator projects such as the ILC. There are collaborative efforts in parallel mesh generation (with SNL), adaptive mesh refinement [48] (with RPI), advanced eigensolvers [39, 45, 49] (with LBNL, Stanford, [59] and UCD), performance improvement (with SNL, LBNL), [62] and shape optimization [61] (with CMU, Columbia, LBNL, LLNL, SNL). Many of these activities have already led to substantial gains both in computational performance and science results while good progress is being made in others.

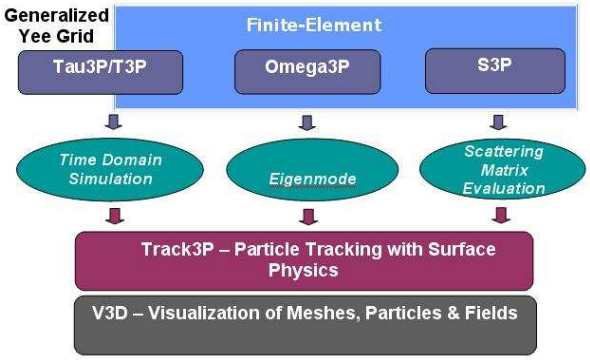

Figure 20: Schematic showing the suite of parallel electromagnetic codes developed at SLAC under the SciDAC Accelerator Simulation project.

5.1.2 Validation of *Omega3P*

Of all the electromagnetic codes developed under SciDAC, ***Omega3P*** is the most matured as its development dates back to the mid 90s when the Accelerator

Grand Challenge was initiated. It saw its first big success [60] under SciDAC in modeling the accelerator structure for the Next Linear Collider (NLC) which was the room temperature technology option pioneered by SLAC for the ILC. Last summer the ILC committee decided on the superconducting technology option championed by DESY which has been accepted and is now being worked on collaboratively by the international accelerator community.

Prior to that decision, ***Omega3P*** was used extensively to prototype the accelerating cells for the NLC's Damped, Detuned Structure (DDS). The DDS is a fully 3D design that is optimized for higher accelerating gradient (14% increase [60] over standard design), and for suppression of harmful wakefields that disrupt long bunch train operation. The computational challenge was to model the complex cell geometry (Fig. 3) to accuracies close to machining tolerance. This is necessary in order to maintain the structure efficiency and to allow for automated machining without manual tuning. It means that an eigenmode solver has to be able to calculate the cavity resonant frequency accurate to 1 part in 10,000 or 0.01%, orders of magnitude beyond what available software running on desktops could provide.

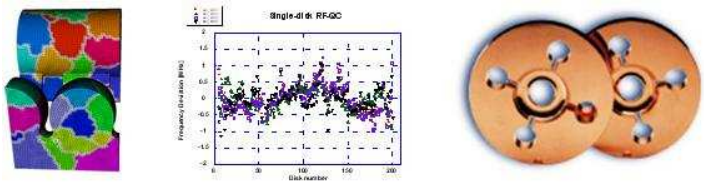

Figure 21: (Left) A distributed ***Omega3P*** model of a quadrant of the DDS, (middle) Microwave QC of the fabricated cells showing measured data within 0.01 percent of target frequency, (right) fabricated DDS cells based on dimensions generated with ***Omega3P***.

Using ***Omega3P*** on the NERSC's Cray T3E, a table of dimensions for all the cells along the DDS was generated for computerized machining based on calculations that met the 0.01 percent frequency accuracy criteria. Cold-test measurements on fabricated cells showed that their resonant frequencies were indeed within 0.01 percent of the target value or 1 MHz deviation from 11.424 GHz. This significant result confirmed ***Omega3P*** as the core design capability for the NLC linac whose DDS design would have saved over one hundred million dollars in machine cost alone.

Figure 22: A model of the 55-cell DDS including input/output fundamental mode couplers and HOM couplers for damping wakefields due to HOMs.

In addition to high resolution modeling of the DDS cell, ***Omega3P*** was

also used in the end-to-end simulation of an entire DDS structure comprising 55 cells together with the power and HOM couplers (see Fig. 4). The computation was to find the dipole wakefields in the DDS by summing the eigenmodes in the frequency domain. To account for the power loss through the fundamental and HOM couplers, a complex solver was developed for ***Omega3P*** to calculate damped modes. A total of 400 complex eigenmodes covering the frequency spectrum of interest was found to form the sum for the wakefields. Fig. 5 shows a comparison between the wakefields from ***Omega3P*** and the result from a time-domain Tau3P simulation using a beam traversing the DDS, showing remarkable agreement [50, 51].

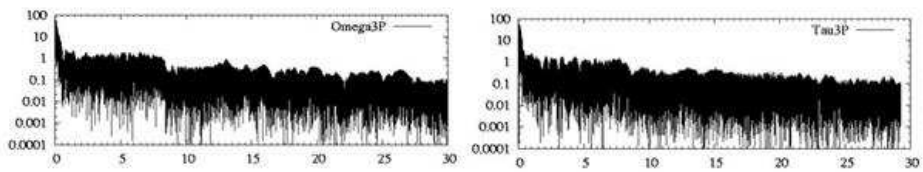

Figure 23: Comparison of DDS wakefields from frequency-domain ***Omega3P*** and time-domain Tau3P simulations.

5.2 Physical Basis for *Omega3P*

The electromagnetic fields in a radio-frequency (RF) cavity are governed by the Maxwell's equations in vacuum. Assuming time harmonic dependence and in the absence of sources, Maxwell's equations can be combined to yield the vector wave equation for the electric field:

$$\nabla \times (\frac{1}{\mu} \nabla \times \mathbf{E}) - \epsilon \kappa^2 \mathbf{E} = 0$$

where ε and μ are the relative permittivity and permeability respectively, and k is the wavenumber which is equal to frequency ω divided by speed of light c.

5.2.1 Lossless Closed Cavity

For an RF cavity enclosed by perfectly conducting walls, the boundary condition at perfectly conducting wall (or electric BC) is

$$n \times \mathbf{E} = 0$$

while the boundary condition (or magnetic BC)

$$n \times \nabla \times \mathbf{E} = 0$$

is applied at a plane of symmetry if one is used. The vector wave equation under these BCs constitutes an eigenvalue problem for the lossless cavity. One

solves for the normal modes of the eigensystem for which the eigenvalues (real) correspond to the normal mode frequencies and the eigenvectors are the normal mode field vectors (real). They form the mode spectrum for the given cavity shape.

To conserve energy, most accelerating cavities are made of copper which can effectively be treated as lossless. Superconducting cavities are more efficient since niobium has even higher conductivity so that all the RF energy can be used to accelerate the beam.

5.2.2 Closed Cavity with Lossy Materials

In the case of an RF cavity with lossy materials within the cavity volume, the relative permittivity becomes a complex number to reflect dissipation in the materials. As a result, the eigensystem is complex which results in complex eigenvalues and eigenvectors. The same BCs apply as in the previous case. The normal mode frequency now has an imaginary part that provides the damping time of the mode.

Lossy materials are used in terminating loads to remove power from the RF cavity and in absorbers strategically located to damp unwanted HOMs in the cavity without affecting the fundamental mode power.

5.2.3 Open Cavity with External Coupling

Power is fed into and taken out of RF cavities via waveguides that are connected to the cavity by couplers. The situation is depicted schematically in Fig. 6 which shows a cavity coupled to three waveguides and the couplers are represented by the gray patches in the cavity-waveguide junctions. This model allows the couplers to be characterized provided the waveguides are properly terminated. For an undriven cavity, one expects only outgoing wave at the waveguide terminating plane. Assuming only one propagating mode in the waveguide at the frequency of interest, the boundary condition is given by

$$n \times \nabla \times \mathbf{E} + i\sqrt{k^2 - k_c^2}\, n \times n \times \mathbf{E} = 0$$

where k_c is the cutoff wavenumber of the waveguide mode. When this so called waveguide BC is added to the electric and magnetic BCs, the eigenvalue problem becomes nonlinear as the eigenvalue also appears there. The solutions to this waveguide loaded cavity are again complex as energy is lost from the cavity through the waveguide terminations. In the actual application, the geometry details of the couplers will be needed.

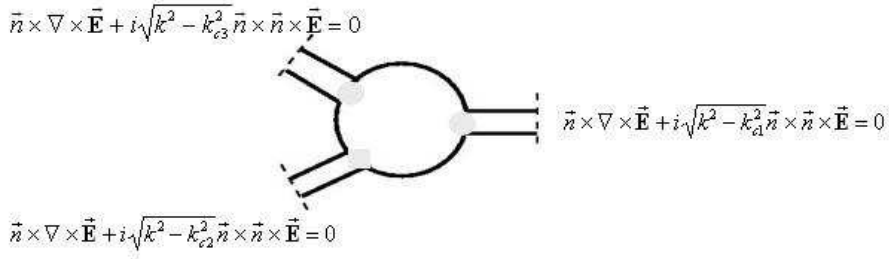

Figure 24: An open cavity coupled to three external waveguides with cutoff wavenumber k_{c1}, k_{c2} and k_{c3}. The couplers are denoted by the gray patches.

5.3 Numerical Basis for *Omega3P*

5.3.1 Finite Element Formulation of Eigenvalue Problem

Using the set of finite-element basis functions $\{N_i\}$ for the electric field, the vector wave equation can be cast into the algebraic eigenvalue problem

$$\mathbf{K}x = k^2\mathbf{M}x$$

where k and x denote the eigenvalue and eigenvector respectively. The stiffness matrix is given by

$$\mathbf{K}_{ij} = \int_\Omega (\nabla\times\mathbf{N}_i)\cdot(\nabla\times\mathbf{N}_j)\,d\Omega$$

and the mass matrix by

$$\mathbf{M}_{ij} = \int_\Omega \mathbf{N}_i\cdot\mathbf{N}_j\,d\Omega$$

In the lossless case, matrix $\mathbf{K}$ and $\mathbf{M}$ are real symmetric while $\mathbf{M}$ is also positive definite. This eigensystem is solved subject to the electric and magnetic BCs imposed.

When external coupling to waveguides is included, the eigensystem becomes

$$\mathbf{K}x + i\sum_j \sqrt{k^2-k_{cj}^2}\,\mathbf{W}_j x = k^2\mathbf{M}x$$

where

$$(\mathbf{W}_j)_{ik} = \int_\Gamma (n\times\mathbf{N}_i)\cdot(n\times\mathbf{N}_k)\,d\Gamma$$

This is the nonlinear eigensystem that was described earlier.

It is worthwhile to point out that H(curl) vector basis functions [41] are used in the finite element formulation to avoid spurious modes in the set of eigensolutions. These so called Nedelec elements [42] can separate the real physical modes with non-zero frequency from the spurious ones that now have

zero frequency. For higher order accuracy, basis functions up to 6th order have been implemented.

5.3.2 Methods of Solution

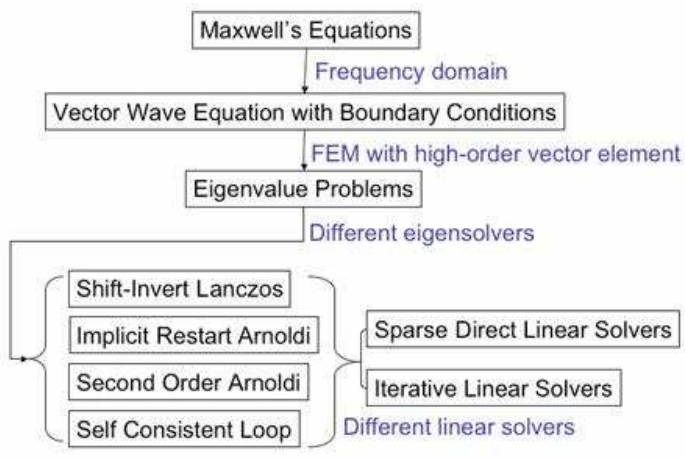

Figure 25: Schematic of eigensolvers and linear solvers implemented in ***Omega3P***

There are four different eigensolvers that are used in ***Omega3P***, depending on the application. For real symmetric eigenvalue problems, shift-invert Lanzos [53] is the method of choice. For complex symmetric eigenvalue problems, shift-invert Arnoldi [52] is used. For complex quadratic eigenvalue problems, second-order Arnoldi [45,54] is preferred. And for complex nonlinear eigenvalue problems, the self-consistent loop [45] is applied.

A linear solver is required for all the eigensolvers and often constitutes the dominant part of the computation time in an ***Omega3P*** run. The list of linear solvers implemented in ***Omega3P*** includes three sparse direct solvers and one iterative solver. The sparse direct solvers are WSMP, [55] MUMPS, [56] SuperLU_dist [57] and while the iterative solver used is Conjugate Gradient with various preconditioners. In terms of speed, WSMP is the fastest linear solver if a large number of eigenpairs are computed. MUMPS is slightly slower but provides a balanced memory load. In terms of memory, Conjugate Gradient requires the least. As a result, WSMP is used over other linear solvers in most Omega3P applications. For larger problems one might choose MUMPS because of WSMP's unbalanced memory requirement. For extremely large problems, Conjugate Gradient with Hierarchical Preconditioner is a better choice as it requires much less memory while providing comparable performance [44].

It is worthy to note that in the sparse direct solvers, the matrix has to be reordered before it is factorized. The best-known matrix-reordering algorithm [58] creates the least fill-in in the factorization stage, therefore requiring the

least amount of memory. It also accounts for a significant amount of execution time for very large problems even though it uses no floating-point operations. Therefore in this case, FLOP count is not an accurate measure of the code performance.

5.3.3 Omega3P Code Description

The latest stable version (v5.1.2) of ***Omega3P*** source code can be found in NERSC's Seaborg under the following directory.

Seaborg: /usr/common/homes/l/llee/OMEGA3P_v5_1_2/

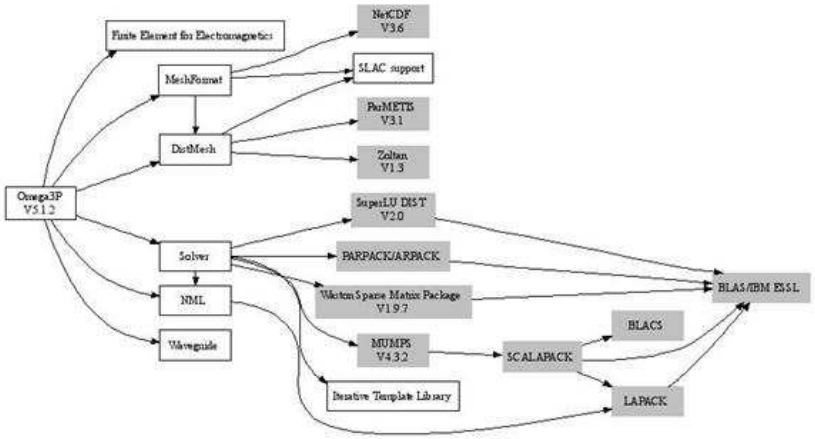

Figure 26: Shematic for library dependency in ***Omega3P***

Omega3P is written in C++ and uses MPI for communication. As shown in Fig 8, the Omega3P application depends on 8 internal libraries (developed and maintained at SLAC) and more than 10 third party libraries. Compiling the code is managed through GNU tools autoconf and make. The configure script can be found at OMEGA3P_v5_1_2/configure. A shell script is provided to run the configure script and set up the path to link third party libraries in Seaborg. This shell script is located at: OMEGA3P_v5_1_2/configure/runConfigureOnNERSC. A compiled executable is also provided at: OMEGA3P_v5_1_2/bin/.

The ***Omega3P*** workflow is shown in Fig. 9. The code reads input parameters from a file. One of the parameters in the input file is a mesh filename with which ***Omega3P*** reads in the mesh and partitions it using either ParMETIS [] or Zoltan []. After that, matrices are assembled according to boundary conditions and finite element basis function order. Then a proper eigensolver is invoked and the requested number of eigenpairs are computed. Omega3P post-processes each eigenpair by computing the electric and magnetic fields on mesh vertices and from these other observables are calculated. Computed field values and observables are saved into files in netcdf format for analysis and visualization. The majority of computation time is spent on the eigensolver and

postprocessing.

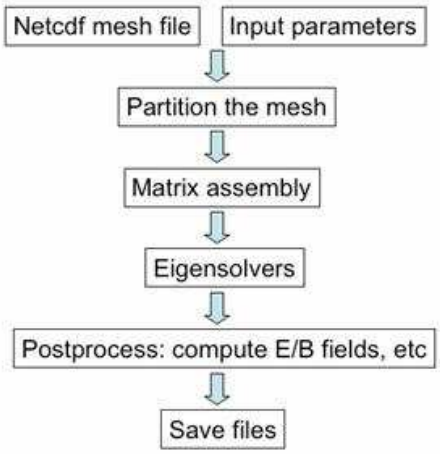

Figure 27: Workflow diagram in ***OmegaP***

5.3.4 Choosing the Hardware Platform for *Omega3P*

The computer allocation for the SciDAC Accelerator Simulation project is mainly on Seaborg at NERSC while time on the ORNL's Cray X1 is also available. To date ***Omega3P*** has been used almost entirely on Seaborg although smaller size problems are run on SLAC's local Linux clusters. As noted previously, the linear solver step takes up the bulk of the computation time so ***Omega3P***'s runtime configuration is mostly determined by the memory requirement for factorizing the matrix. WSMP has been the solver of choice because of its superior speed. Since WSMP is written in mixed-mode programming (thread and MPI), ***Omega3P*** runs on Seaborg with different configurations of number of threads per task (MPI process) and number of tasks per node, depending on the memory requirement of the application. Because ***Omega3P*** is solely MPI-based, numerical experiments show that the optimal configuration is 4 tasks per node with 4 threads per task. In case of large problems, 1 task per node with 16 threads per task is also used.

5.4 *Omega3P* Science Cases

For this report, the focus is on the application of ***Omega3P*** to assess the HOM damping in the superconducting accelerating cavities for the ILC. In the first quarter, the computations were performed on the TTF design with the attention turning entirely to the LL design in the second quarter.

5.4.1 Omega3P Input Parameters

Omega3P requires a set of input parameters which are organized into groups and put in a file. The input parameters are:

1. ***Geometry model input*** - Tetrahedral mesh file in netcdf format; boundary condition specifications on surfaces; waveguide port location and dimensions with number of loaded modes included.

2. ***Discretization input*** - Order of finite element bases and order of geometry curvature used.

3. ***Eigensolver input*** - Form of eigenvalue problem: linear, quadratic or nonlinear; choice of solvers: Second-Order Arnoldi (SOAR), Self-Consistent-Loop (SCL), or Exact Shift-Invert Lanczos (ESIL); choice of linear solvers: WSMP, MUMPS, SuperLU_dist, or CG; frequency shift for range of eigenvalues of interest; and number of eigenvalues desired.

4. ***Postprocessing input*** - Specifications for plotting Electric and Magnetic field arrows, contours or lines; specifications for calculation of output observables (list follows).

5.4.2 Omega3p Output Observables

Omega3P output observables include:

1. Resonance frequencies of the cavity

2. Electric and Magnetic fields in the cavity

3. Wall loss due to dissipation on the metal surface

4. Quality factor

5. R/Q

6. Shunt impedance (kick factor)

7. External quality factor

8. Number of DOFs used in the simulation, number of nonzeros in the matrices, and the partitioning in different processors.

9. Timing results for mesh reading, matrix assembly, eigensolver, postprocessing, and file writing.

5.5 Q1: TTF Cavity for the International Linear Collider

Following the ILC decision on the superconducting technology option, the modeling effort at SLAC has reoriented towards the TTF design of the superconducting RF cavity. While the design has been in existence for over a decade, no simulation had been done on the cavity in its entirety because both the tool and the resources were not available. As mentioned earlier, understanding and quantifying the HOMs in this cavity are of utmost importance to maintaining beam stability in the ILC. In the first quarter, SLAC achieved the first end-to-end simulation of the TTF [38] cavity and has since been designated as the computational group to support the ILC accelerating cavity R and D. Presently, an international team comprised DESY (Germany), KEK (Japan), Jefferson Lab and SLAC (US) is working on an improvement to the TTF design called the Low-loss (LL) design. The LL cavity has been the focus of the SLAC team in the second quarter. A model of the original TTF cavity is shown in Fig, 10.

It is important to point out that preprocessing and postprocessing constitute a significant portion of the work for an ***Omega3P*** simulation. Preprocessing includes preparing the geometry model from CAD drawings and meshing. It is essential to perform a convergence study as a function of mesh sizes to ensure that the mesh accurately describes the geometry in the physics regime of interest. In addition to the derivation of output observables in the postprocessing step, visualization is absolutely vital in understanding the numerical results and a substantial amount of work has been devoted to the development of graphics tools for plotting 3D field data from unstructured grids.

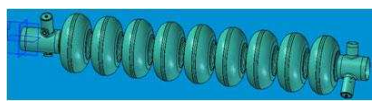

Figure 28: Model of the TTF cavity for the ILC

5.5.1 Physics Output

Based on CAD drawings from DESY, the TTF cavity for the ILC was modeled for the *first time* ever using Omega3P. The fundamental mode was first found to check the accelerating mode frequency and the accelerating field flatness (Fig. 11). The first two dipole bands were then computed to find their RF properties of which the Qext has the most significance as it determines how persistent the HOM is in affecting the beam. Qext is calculated from the complex eigenfrequency which is the solution to the nonlinear eigenvalue problem when the waveguide BC is considered at all the coupler ports. Comparison of the results for the first dipole band from Omega3P and measured data shows extremely good agreement (Fig. 12). The simulation also provides details of field distribution in the coupler regions which are extremely difficult to obtain otherwise, and are invaluable for engineering and design improvement purposes

(Fig. 13). Preliminary results were presented at the first ILC workshop in KEK in November of last year and attracted a lot of attention.

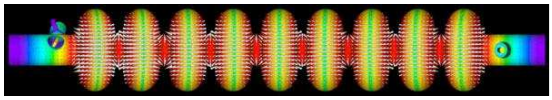

Figure 29: Electric field distribution of the accelerating mode in the TTF cavity.

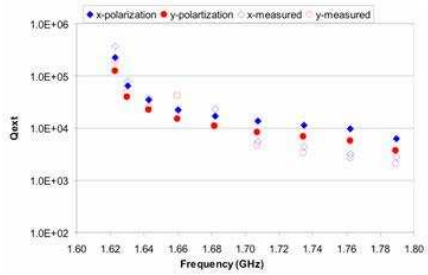

Figure 30: Q_{ext} results between Omega3P and measurements for the two polarizations of the eigenmodes in the first dipole band of the TTF cavity for the ILC.

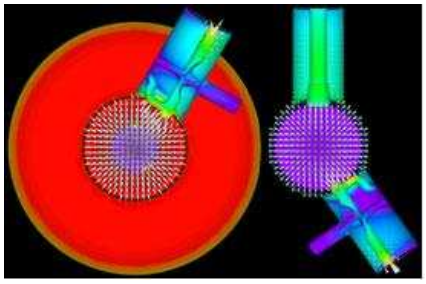

Figure 31: Electric field distribution of a HOM in the input (left) and output (right) end of the TTF cavity, showing details of the coupling loops.

5.5.2 Algorithmic Implementation

In this quarter, the Self-Consistent-Loop (SCL) algorithm [45] was implemented in Omega3P to solve the nonlinear eigenvalue problem required for the TTF cavity modeling.

5.5.3 Computational Performance

The TTF cavity model consisted of 742K tetrahedral mesh elements that translate to 816337 degrees of freedom (DOF) when linear basis functions are used. With the SCL solver, each eigenmode took about 315 seconds to compute on 256 CPUs of Seaborg and required a total of 60GB memory. These numbers will serve as the baseline for future computations the ILC cavity.

5.6 Q2: Low-loss Cavity for the International Linear Collider

The International Linear Collider community plans to finalize the baseline design for the accelerator by this August when the 2nd ILC Workshop will convene during the Snowmass meeting. The Low-loss (LL) accelerating cavity design is a potential improvement (e.g. 20 percent less cryogenic loss) to the existing TTF prototype. Currently, SLAC is working with DESY, KEK and Jlab on two variants of the LL design to predict their performance computationally. Although both DESY and KEK are on schedule in building a prototype, it is evident that only simulation will be able to provide the input for determining the baseline cavity design for the ILC by Snowmass.

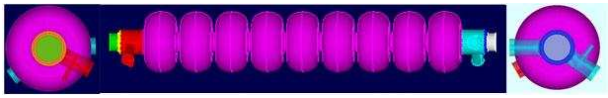

Figure 32: Model of the TTF cavity for the ILC

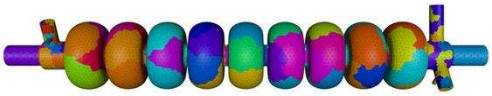

Figure 33: Distributed mesh of the ILC LL cavity partitioned with ParMETIS.

5.6.1 Physics Output

In the second quarter, SLAC devoted considerable resources to simulating the Low-Loss cavity design due to the urgency in the ILC time schedule. Using a new eigensolver, the first four monopole bands and the first five dipole bands were computed, resulting in almost five times as much in physics output (Fig. 14).

5.6.2 Algorithmic Implementation

A newly developed algorithm for solving quadratic eigenvalue problems, the Second-Order Arnoldi (SOAR), was implemented that led to significant improvement in solver efficiency and speed [45].

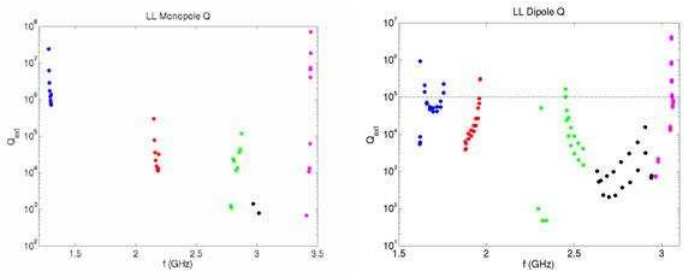

Figure 34: External Q results for 4 monopole bands(left) and 5 dipole bands (right) in the LL cavity design for the IL

5.6.3 Computational Performance

A vigorous convergence study resulted in an optimal mesh with 523K tetrahedral elements. Using quadratic basis functions with second-order geometry, the overall number of DOFs is 3182010, which is about 4 times that of the TTF cavity. Applying the SOAR algorithm, it took about 5560 seconds to compute 18 modes using 768 CPUs and 300GB memory on Seaborg. This is a factor of 5 increase in problem size for the same computation time. Approximately 200 eigenmodes have been found for this new cavity design compared to the 40 calculated for the TTF cavity.

5.7 Q3 Progress and Dynamics

The new development in the global ILC cavity design effort is the ICHIRO cavity proposed by KEK to be another viable high gradient option to the existing TTF cavity. KEK is planning to fabricate and test this cavity with a target goal of 50 MV/m. The ICHIRO cavity is based on the LL design but has a larger beampipe to fit into a multi-cavity superstructure. KEK plans to build a 4 to 8 cavity cryomodule in their Superconducting Test facility in two years. SLAC is providing the main simulation support to this project.

At the same time, SLACs computational design of the ILC cavity was one of four projects selected to receive significant time allocation on the Cray-X1 at ORNL called Phoenx. Considerable effort was then devoted to porting Omega3P to Phoenix. In initial testing, it was identified that the direct solver MUMPS was taking excessively long time to factorize a moderate-size sparse matrix because of matrix-reordering using METIS. METIS is a graph-based algorithm which often cannot be efficiently muti-streamed or vectorized. As a result, we implemented a customized matrix re-ordering interface to access the parallel nested-dissection algorithm in ParMETIS which significantly reduced the execution time for matrix reordering in the Omega3P runs. Table[10] shows the comparison of execution time and memory used for two matrix reordering algorithms. For this example, the number of non-zeros in the sparse matrix is 64293136. With ParMETISs nested-dissection algorithm, the total execution

time decreased from 1368 seconds to 1038 seconds.

		METIS	ParMETIS
Analysis	Matrix Reordering	517 seconds	99 seconds
	Symbolic Factorization	200 seconds	201 seconds
	Numeric Factorization	165 seconds	192 seconds
	Time for Eigensolver	1368 seconds	1038 seconds
	Memory Used for Factorization	46 GB	65 GB

Table 10: Execution time and memory used for a moderate-size quadratic eigenvalue problem solved with SOAR/MUMPS on 32 MSPs of the Cray X1 at ORNL.

In production runs on Phoenix, Omega3P computed 18 modes in 5000 seconds using 32 MSPs and about 200 GB memory. This quadratic eigenvalue problem has a matrix size of 3516514 and was solved on 32 nodes to fulfill the large memory requirement.

Using the optimized Omega3P on Phoenix, we simulated the HOM damping in the ICHIRO cavity. Because of the larger beampipes, the dipole modes that are trapped inside the cavity are limited only to the first two bands. The results are shown in Figure[35]. As can be seen, a mode in the first band exhibits undesirably high Q so work is in progress to redesign the cavity so that the Q can be lowered to within specified limit. Many of the initial runs on this cavity to test for mesh convergence were carried out on Seaborg at NERSC.

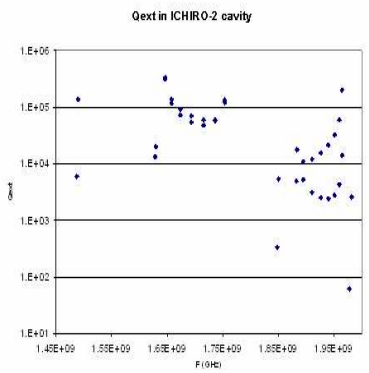

Figure 35:

In the ICHIRO cavity, the HOMs that are higher in frequency than the first two bands we calculated can now propagate out of the cavity through the larger beampipes. To study the damping of these modes above the beampipe cutoff, a two cavity model is needed and ultimately the whole chain of cavities in a cryomodule will have to be simulated. A model of the STF 4-cavity test

cryomodule is shown below in Figure[36]. Such problems will be much larger in size and will require further advances from the eigensolvers in Omega3P. Efforts in this direction have just begun.

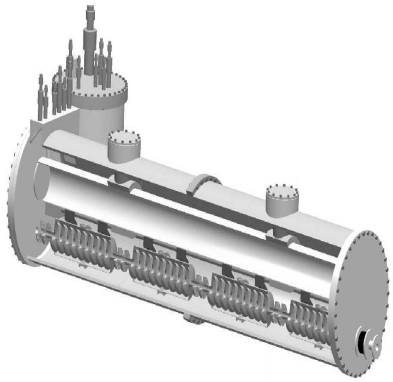

Figure 36:

5.8 Q4 Progress and Dynamics

5.8.1 Super Structure (SST) for the International Linear Collider

The International Linear Collider has formed the Global Design Effort (GDE) during the Snowmass meeting in August and one of its goals is to improve the performance and lower the cost of the machine. One third of the cost comes from the accelerating system consisting of the superconducting cavities that are being optimized through experimentation and simulation. One candidate under investigation is the Super Structure (SST) proposed by DESY which comprises two 9-cell cavities therefore requiring one less power input coupler. A configuration of the SST is shown in Figure[37].

5.8.2 Physics Output

Of concern in this design are the HOMs trapped between the cavities and whether the HOM couplers are sufficiently effective in damping these modes. Omega3P was used to calculate the Qext of the trapped modes, some of which are shown in Figure[38]. Work is ongoing to optimize the HOM couplers and end cell geometries to reduce the Qs of the highly trapped modes that are not damped sufficiently.

5.8.3 Algorithmic Implementation

In this quarter, two new algorithms based on the Nonlinear Arnoldi and Nonlinear Jacobi-Davidson methods have been implemented into Omega3P. Their

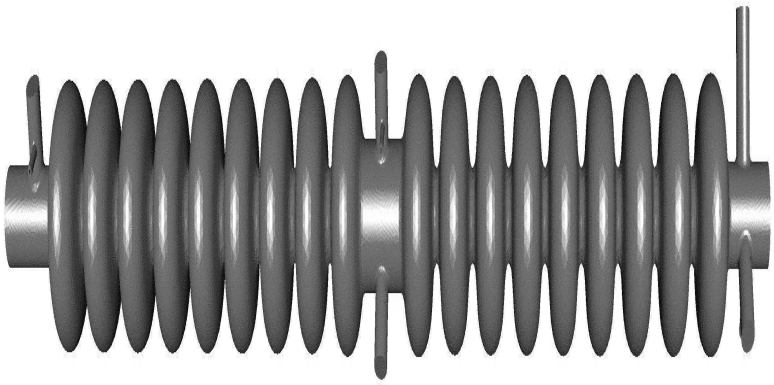

Figure 37: CAD model of the SST design for International Linear Collider.

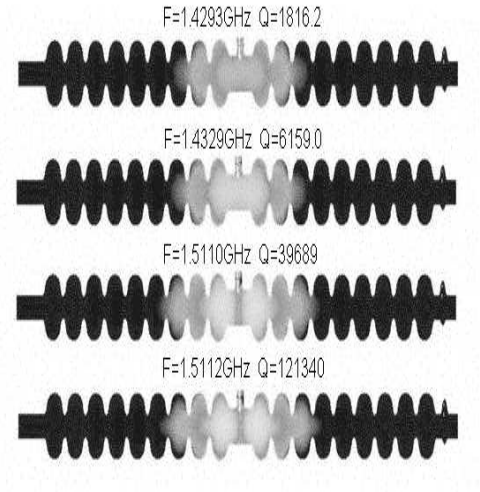

Figure 38: Mode patterns of trapped dipole modes in the SST with respective frequency and external quality factor.

application is aimed at solving for the high Q modes in a cavity that is coupled to several waveguides each with different waveguide cutoff frequencies. In this case, the nonlinear eigenvalue problem cannot be reduced to the quadratic eigenvalue problem (QEP) case so therefore special eigensolvers are required.

5.8.4 Computational Performance

During this quarter, great improvement in the performance of the parallel Second Order Arnoldi (SOAR) algorithm in Omega3P has been made through the SciDAC collaboration with the PERC ISIC. The run-time performance analysis was generated by LBL which provided the insight that SLAC researchers used to optimize Omega3P for solving the quadratic eigenvalue problem. When applied to the Low-Loss cavity modeled by 523K tetrahedral quadratic elements, the performance comparison for computing 18 eigenpairs on Phoenix (X1E) and on Seaborg (IBM SP) is shown in Table[11]. Relative to Q3 results, Q4 performance is more than 30% higher in terms of executing time.

Eigensolver Times	Pheonix	Seaborg
Q3	2738 sec	3339 sec
Q4	1869 sec	2114 sec

Table 11: On Pheonix 32 MSPs were used to calculate 18 eigenpairs. On Seaborg 32 nodes were used to compute the 18 eigenpairs. There were 2 tasks per node and 8 threads per task on the IBM runs.

6 LAMMPS -a classical molecular dynamics package

6.1 Overview of Molecular Dynamics

Classical molecular dynamics (MD) is a simulation technique for modeling the atomistic properties of solids, molecules, and solutions. Point-mass particles represent atoms, molecules, or coarse-grain objects which interact with each other via analytic formulas that represent the physics incorporated in the MD model. Typical formulas include pairwise VanderWaals and Coulombic interactions and 2-, 3-, and 4-body bond-stretch, bond-bend, and torsional interactions in molecular systems. MD simulations are typically deterministic; atoms are assigned initial velocities and Newton's equations of motion are integrated forward in time. From the trajectories of the ensemble of atoms various thermodynamic, structural, and kinetic information about the system can be inferred. Depending on the initial and boundary conditions applied to the simulation, equilibrium or non-equilibrium phenomena can be modeled. Prototypical applications of MD include protein and lipid bilayer conformational studies, phase diagram de-

termination for novel materials, and the response of solids to shock, stress, or thermal cycling.

Computationally, MD is similar to the N-body problem. Unlike gravitational or plasma simulations, the forces in MD are mostly short-range and particle densitites do not reach high values. Typically, the breakdown of CPU cost for a timestep is 85% for force computation, 10% for neighbor finding, and an additional 5% which includes time integration, application of boundary conditions, etc. The force computation is dominated by short-range pairwise interactions, which is why efficient neighbor finding is critical. VanderWaals interactions are typically cutoff at a distance which encompasses from a dozen to several hundred neighbors. If Coulombic interactions are modeled, long-range interactions are often included for accuracy. These are split into a short-range direct portion similar in extent to the VanderWaals interactions and a long-range K-space portion which is computed by Ewald summation. The most efficient methods for this summation are solutions to Poisson's equation via 3d FFTs on a grid to which particle charge is interpolated. Ignoring the $O(NlogN)$ cost of FFTs (which typically only require 20-30% of the force computation time), classical MD simulations scale as $O(N)$ in both memory and CPU cost, where N is the number of particles simulated. They also parallelize efficiently, at least for large problems, with typical parallel efficiencies of 80-90% on thousands of processors for simulations with millions of atoms.

The timestep in an MD simulation is limited by the need to accurately integrate atomic motion between strongly interacting atoms, e.g. between 2 atoms coupled by a harmonic bond. For all-atom models of proteins or polymers this requires a timestep of about a femtosecond (for coarse-grain models, it can be a few orders of magnitude larger). The current state-of-the-art for supercomputer-scale simulations is that tens of nanoseconds (tens of millions of timesteps) can be simulated for models with tens to hundreds of thousands of atoms. This requires many hours or days of CPU time on hundreds of processors of a parallel machine. Similary, for solid state systems, tens of millions of atoms can be simulated for shorter timescales. Note that this still implies a significant length-scale limitation since there are a few billion atoms in a cubic micron of solid material. Because of their computational intensity, such problems are good stress tests of the performance and scalability of large parallel machines.

6.2 Overview of LAMMPS

LAMMPS, which stands for Large-scale Atomic/Molecular Massively Parallel Simulator, is a classical molecular dynamics code developed primarily at Sandia National Laboratories over the last 10 years with various kinds of DOE support. Initially, it began as a Cooperative Research and Development Agreement (CRADA) between Sandia, Lawrence Livermore Naional Laboratories (LLNL), and 3 companies (Cray Research, Bristol-Myers Squibb, and Dupont). This effort resulted in a Fortran code that was massively parallel and distributed by a no-fee license agreement.

Following the 3-year CRADA, development continued with smaller amounts

of support from various DOE-funded projects (LDRD, ASCI, BES) at Sandia, as the code was enhanced to apply it to new kinds of models. For the last 3 years, we have had support from DOE's OASCR and OBER Genomes-to-Life project [32] to apply LAMMPS to specific protein-modeling problems. Additionally, we have had support from an LDRD focused on coarse-grain lipid membrane models and from a BES grant focused on granular material modeling.

The union of these 3 projects has enabled us to merge the capabilities of several of our parallel MD codes into the current LAMMPS, which was re-written as an object-oriented C++ code so as to enable a wide variety of MD material models, force fields, boundary conditions, and other features to be incorporated in a modular fashion within a single code. The new code is designed to be easy to extend by users and is now distributed as open-source code under the GPL licence. It can be downloaded from the LAMMPS WWW site [33]; since the initial public release in September 2004, there have been about 3200 downloads. The LAMMPS WWW site has extensive documentation (175-page user manual), example calculations and visualizations (snapshots, movies), as well as a list of about 60 publications that have resulted from simulations performed with LAMMPS and its predecessor codes.

LAMMPS has options for the following kinds of simulations. Some of these features were added in FY05 and are discussed further below:

- systems: atomic, polymeric, biological, metallic, granular
- pairwise potentials: Lennard-Jones, Coulombic, frictional granular
- molecular potentials: bond, angle, dihedral, improper
- polymer potentials: all-atom, united-atom, bead-spring
- charge: Ewald, particle-particle particle-mesh (PPPM), dipolar
- CHARMM and AMBER force-field compatability
- constant NVE, NVT, NPT integrators
- rRESPA hierarchical timestepping
- SHAKE bond and angle constraints
- parallel tempering (replica exchange)
- targeted molecular dynamics (TMD) constraints
- run one or more simulations simultaneously in parallel
- wide variety of boundary conditions and constraints

LAMMPS runs on single-processor machines or in parallel using message-passing techniques and a spatial-decomposition of the simulation domain [30,31]. Because it uses the standard MPI library for message passing, it is portable to

virtually every available parallel platform. The only additional library it needs is for 1d FFTs (used to compute the 3d FFTs needed for the PPPM solver); users can link to vendor-supplied libraries on specific machines, or use the freely available FFTW library. For computational efficiency, LAMMPS computes and stores Verlet neighbor lists of pairwise interactions by $O(N)$ binning techniques.

6.3 Code Development Plans for FY05

In this section we highlight the various code improvements we plan to add to LAMMPS during FY05. Some of these features are in response to user requests, others are to support the science projects discussed in the next section.

6.3.1 Algorithmic Improvements

- Allow groups of atoms (molecules, protein residues, etc) to be treated as rigid bodies acted on by forces and torques for the purpose of faster time integration and longer timesteps.
- Implement targeted molecular dynamics (TMD) as a constraint method to enable a group of atoms to be driven from one conformation to another over the course of a simulation. This enables conformational changes to be more rapidly simulated and the free energy of a barrier transition to be computed.
- Implement Monte Carlo bond-swapping for coarse-grain polymer melts. This enables much faster equilibration of polymer systems.

6.3.2 Speed Enhancements

- Allow the option to pre-tabulate short-range Coulombic forces for use with the long-range Ewald and PPPM solvers.
- Enhace the rRESPA hierarchical time stepping capability to partition pairwise force computations by distance: inner, middle, outer. This should enable longer timesteps for costly long-range Coulombic computations.
- Implement native 1d FFT support for SGI Altix platforms.

6.3.3 New Features

- Implement hybrid atom styles and pair and bond potentials. These will enable more complex systems to be simulated, e.g. a polymer or biomolecule on a metal surface or a dipolar solvent with an all-atom protein.
- Implement a generalized pairwise force field whose values for energy and force as a function of distance are read-in from a pre-tabulated file. This will allow rapid experiementation with a variety of novel force fields without the need to formally implement them in LAMMPS.

- Implement one or more energy minimizers based on steepest descent or conjugate gradient algorithms. These are useful for determining structures of molecular systems or quenching kinetic energy from liquid or solid-state systems.

- Implement so-called class 2 inter-molecular potentials. These are thought to be more accurate formulations of bond-stretch, bond-bend, and torsional-twist interactions for some kinds of molecular systems. They include higher-order terms in their analytic formulations.

- Enable computation and dumping of per-atom energy, stress, and the centro-symmetry parameter. The latter is a useful measure of local order and defect characteristics in a solid.

- Implement 3-body force fields for semiconductor materials (e.g. Stillinger-Weber or Tersoff potentials for silicon-based materials).

- Allow for specification of particles with point-dipoles, including the charge-dipole and dipole-dipole force fields associated with such systems. This will enable more realistic simulation of coarse-grained biological systems where solvents or charged atom groups are treated as point dipoles.

- Implement and release a pre- and post-processing package for LAMMPS written in the scripting language Python. It will allow for hi-level setup of LAMMPS simulations (e.g. solvation of proteins, assignment of force field parameters, visualization of systems as they are built, GUI-driven setup of LAMMPS input files, etc) as well as post-processing of LAMMPS output (reading and statistical computations on thermodynamic outputs and dumped snapshots of atom configurations, conversion to different formats, plotting of relevant quantities, visualization and movie-making from snapshot files, etc).

On the last point, LAMMPS has been designed as a fast, lightweight parallel engine for molecular dynamics (MD) simulations. As such, the LAMMPS kernel provides only a modest amount of support for setting up simulations and analyzing their output. We anticipate this new pre- and post-processing package will be of great utility to LAMMPS users who do not have the time or inclination to write their own tools for such purposes. We do not intend to re-invent the capability available in other visualization or plotting packages, but rather to use Python as a wrapper on other tools so as to make them easily accessible to LAMMPS users within a standardized, scriptable interface.

6.4 Science Plans for FY05

Since LAMMPS is an open-source code that has been downloaded over 2500 times, we do not know all the simulations people are performing with LAMMPS. In this section we highlight 3 simulation projects that we and our close collaborators are using the code for during this fiscal year. These DOE-sponsored

projects span a spectrum of material models and LAMMPS capabilties and are also motivating some of the FY05 enhancements to LAMMPS, as outlined in the previous section.

6.4.1 RuBisCO protein

We are studying the protein RuBisCO using LAMMPS and the targeted molecular dynamics (TMD) method to show how structural changes in the binding niche gating mechanism alter the overall enzyme specificity and performance. RuBisCO is the primary carbon fixation enzyme in plants and bacteria, including the cyanobacteria Synechococcus, which has been a focus of our DOE-funded genomes-to-life (GTL) project. Carbon fixation is the process by which inorganic forms of C are converted to organic sugars, a process DOE is interested in because such organisms sequester Carbon and play a key role in the global carbon cycle which in turn affects global warming.

From an enzymatic viewpoint, RuBisCO's poor specificity and inefficiency represent a bottleneck in carbon fixation and the photosynthetic process. In silico mutations of residues in the C-terminal region affect the computed free energy barrier for the gating mechanism. This computed free energy barrier can be used to predict the performance of a given RuBisCO structure.

Using LAMMPS, we have predicted the gating barriers for Synechococcus, rice, tobacco, and spinach RuBisCOs, as well as other mutant forms. As expected, higher specificity rice and tobacco forms of RuBisCO have smaller gating barriers than wild-type Synechococcus RuBisCO. Also, destruction of a salt bridge between the C-terminus and the protein wall by D473A mutation reduces the gating barrier. Continued TMD simulations using LAMMPS are enabling discrimination between the RuBisCO forms and are helping rationalize the effects of structural variations on enzyme performance, as in Figure 6.4.1. A submission of this work to the Biophysical Journal is listed below [35].

6.4.2 Metal Islands on Silica

To investigate fundamental mechanisms of stress generation during thin film growth, we are performing large-scale atomistic simulations using LAMMPS, as part of a DOE-funded LDRD project at Sandia. Residual stress in thin film structures is known to dramatically influence device performance. To address this, we are performing simulations of islands on substrates where the island is metal and the substrate is either metal or oxide. These combinations were selected as they embody a broad range of material properties that may be encountered in thin film processing.

For metal-on-metal simulations, embedded atom method (EAM) potentials can be used. This capability was recently incorporated into LAMMPS, providing a significant performance improvement over the earlier simulation code used. For metal/oxide combinations, we plan to use EAM potentials for metal/metal interactions, a Buckingham potential in combination with Coulomb forces for oxide/oxide interactions, and a Lennard-Jones (12-6) potential for the metal/oxide

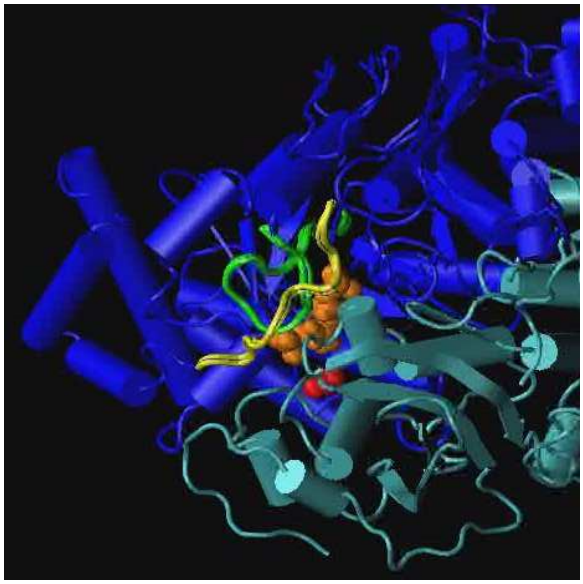

Figure 39: Snapshot of a LAMMPS simulation of a solvated RuBisCO protein and its binding pocket. The two halves of the protein dimer appear in blue and gray; different portions of the binding pocket gate are in yellow and green; the ribulose substate ligand is in orange, a CO_2 molecule is in red; surrounding water has been removed from the image.

interactions. This permits arbitrary control over the strength of interaction between island and substrate (i.e. island adhesion). The capability to combine hybrid potentials in a single simulation is a new feature being added to LAMMPS in FY05. With this tool, isolated islands of varying size will be simulated to extract the dependence of stress in the island on island size and adhesion to the substrate. In addition, when islands grow large enough to impinge upon one another, island coalescence is theorized to be a primary contributor to stress generation. Thus far, coalescence for islands with radii of 100 nm has been studied using simulations of a few million atoms. Future simulations will examine hemispherical islands, requiring tens of millions of atoms for the largest sizes we hope to explore.

6.4.3 Slow Granular Flows

Slow flows in dense granular materials are of practical interest in a variety of applications such as soil failure and powder processing, and as of yet are poorly understood. While progress has been made in understanding rapid flows in terms of kinetic theory, the nature of slow flows limits the use of such approaches. These denser systems often exhibit thin, localized regions of particle motion,

shear bands, separating largely solid-like, immobile regions. To understand the flow in dense granular materials, we are carrying out a combined experimental and simulation study of flow in a split-bottom Couette cell. Unlike traditional Couette cells with an inner wall that is rotated, an inner disk on the bottom of the cell is used to shear the system. This eliminates the need for an inner wall which can have a strong effect on the size of the shear zone.

Using LAMMPS, it is now possible to model the same system sizes as in the experiment (50-180 thousand particles for 80 million timesteps). As the fill height of the particles in the Couette cell increases, the width of the shear band increases and moves away from the outer wall. Results for azimuthal surface and depth velocity in piles of various heights from both experiments using glass beads and discrete element simulations have been gathered as in Figure 6.4.3. The data can be rescaled to fall on a universal curve regardless of the particle properties. This universal curve reveals excellent agreement between experiment and the simulation. However, in contrast to previous theoretical descriptions, we found that when the height of the pack reaches a specific value a transition occurs at which slip between layers becomes increasingly significant; giving rise to an additional vertical shear band. We are presently working with theorists to examine the simulation results in light of a recent theoretical description developed for quasi-static granular flows based on an analogy with solid friction where layers of particles are modeled as "shear-free sheets" that slide past one another. A submission of this work to Physical Review Letters is listed below [36].

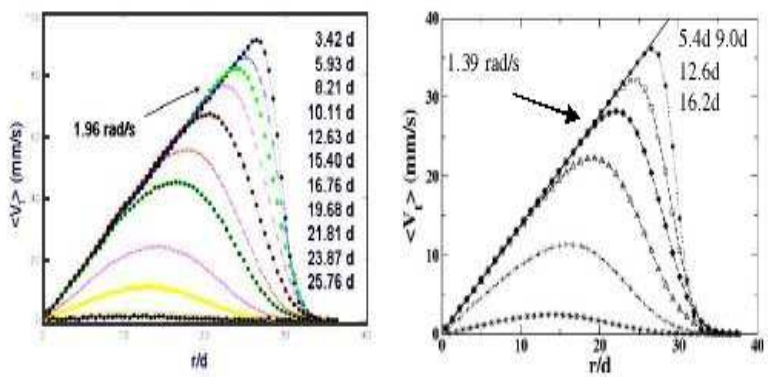

Figure 40: Azimuthal surface velocity profiles for packs of varying height in terms of particle diameter d. Experimental results for glass beads (left) and simulation (right) are shown.

6.5 Benchmarking Effort for the Joule Project

In this section we describe benchmark calculations performed with the LAMMPS MD code and which are being tested on various platforms during FY05 as part of the Joule project. The benchmark section of the LAMMPS WWW page [33] lists 5 different classes of problems; here we focus on the 3 that are most relevant to the science objectives described above: a solvated protein embedded in a lipid bilayer, a solid-state bulk metal, and a flow of granular particles.

For benchmarking purposes, a model for each of the 3 styles was created with 32,000 atoms (or particles in the case of the granular system), a size typical of small MD problems users might run on a desktop or for very long timescales on a parallel machine. Each problem can then be replicated within LAMMPS to generate larger versions.

The listed CPU times are for benchmark runs of 100 timesteps. The bulk of the cost differences between the 3 benchmarks is due to the expense of computing a particular pairwise force field for a given cutoff distance which affects the number of neighbors per atom. Since classical MD scales linearly with the number of timesteps and atoms simulated, these results can easily be used to estimate the CPU time a long simulation (millions of timesteps) of a given style would require on a given number of processors.

For each of the benchmarks, a plot of fixed- and scaled-size timings is shown below. Fixed-size (strong scaling) means that the same 32K atom problem was run on varying numbers of processors. Scaled-size (weak scaling) means that when run on P processors, the number of atoms in the simulation was P times larger than the one-processor run. E.g. a 64-processor scaled timing is for a run with 2,048,000 atoms.

The plotted parallel efficiencies refer to the ratio of ideal to actual run time. For example, if perfect speed-up would have given a run-time of 10 seconds, and the actual run time was 12 seconds, then the efficiency is 10/12 or 83.3%. The parallel runs were typically made on production machines while other jobs were running, which can sometimes degrade performance.

Each of the plots in Figures 41-43 shows scaling data for 7 machines. Note that the efficiency curve for each machine is scaled by the one-processor CPU timing on that platform (shown in parenthesis in the plots). Thus, for the same number of processors, a fast-CPU machine running at 60% efficiency may run the benchmark in a wall-clock time much faster than a slow-CPU machine running at 90% efficiency.

The 7 machines benchmarked are as follows:

- ASCI Red = 9000 333MHz Intel Pentium 3 processors with a custom Intel inteconnect.
- Ross = 1000 500MHz DEC Alpha processors with a Myrinet interconnect.
- Liberty = 472 3.0 GHz Intel 32-bit Xeon processors with a Myrinet interconnect.

- Spirit = 1024 3.4 GHz Intel 64-bit Xeon processors with a Myrinet interconnect.
- Cheetah = 864 1.3 GHz IBM Power4 processors with an IBM Federation interconnect.
- HPCx = 1600 1.7 GHz IBM Power4+ processors with an IBM Federation interconnect.
- Blue Gene Light = 65536 700 MHz PowerPC 440 processors with a custom IBM interconnect.

The first 4 machines in this list are sited at Sandia National Labs (SNL); cheetah is at ORNL, HPCx is at the Daresbury Lab in the UK, and Blue Gene Light is at LLNL. The ross, liberty, and spirit machines are prototypical Linux-based clusters built with commodity processors and interconnect hardware; the other machines are traditional tightly-coupled massively parallel machines with custom interconnects.

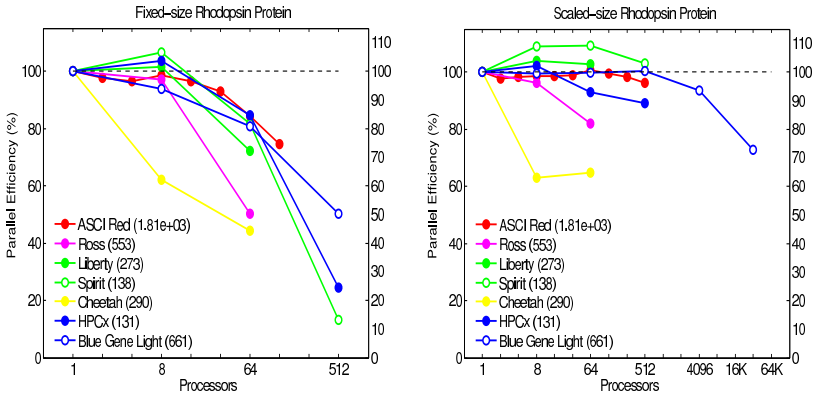

Figure 41: Parallel efficiency of LAMMPS on 7 machines for the rhodopsin protein benchmark. Fixed-size efficiency (left) and scaled-size (right) are shown. The one-processor timing for the simulation on each machine is shown in parenthesis.

A few additional details about the 3 benchmark problems are listed here. More details and data on the performance results are discussed on the benchmark section of the LAMMPS WWW page [33].

- Protein benchmark: All-atom rhodopsin protein in solvated lipid bilayer with CHARMM force field, long-range Coulombics via PPPM (particle-particle particle mesh), SHAKE constraints. This model contains counter-ions and a reduced amount of water to make a 32K atom system. 10 Å cutoff for LJ interactions; 440 neighbors per atom.

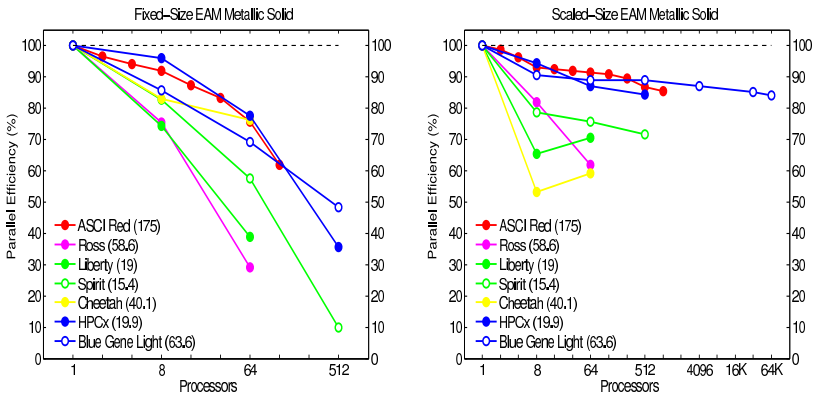

Figure 42: Parallel efficiency of LAMMPS for the metallic solid benchmark. Same format as the previous plot.

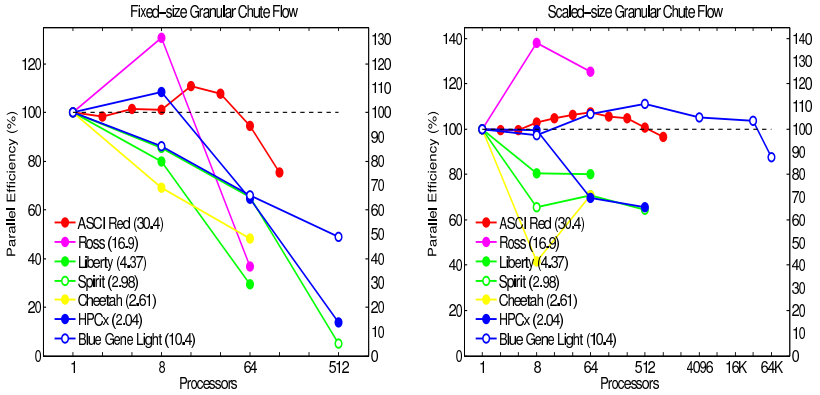

Figure 43: Parallel efficiency for the granular chute flow benchmark. Same format as the previous plot.

- Metal benchmark: Cu metallic solid with embedded atom method (EAM) potential. 4.95 Å cutoff; 45 neighbors per atom.
- Granular benchmark: Chute flow of packed granular particles with frictional history potential; 7 neighbors per atom.

Additionally, a benchmark of an atomistic Lennard-Jones fluid problem has been benchmarked for very large systems to test the scalability of LAMMPS on tens of 1000s of processors. The LJ calculation is the kernel pairwise force interaction common to all the benchmarks discussed above. It was run with 40 billion atoms on on 65536 processors of the new LLNL Blue Gene Light) machine. It ran in 585 seconds for 100 timesteps for a parallel efficiency of about 92%.

The aggregate flop rate for this run was 4.34 Tflops, using these values (conservatively counted):

- 40×10^9 atoms
- 27.6 pairwise interactions per atom (with Newton's 3rd law for this density and cutoff)
- 23 flops per LJ interaction
- 5.85 secs per timestep

6.6 Q1 Progress and Dynamics

The following tasks were worked on in Q1 (Oct-Dec) of FY05 as listed in the goals outlined above.

The inital public open-source LAMMPS release was made on 1 Sept 2004.

Algorithmic improvements: The tasks for rigid body dynamics and targeted molecular dynamics were implemented. These were important for the RuBisCO modeling described above.

Speed enhancements: The tasks for rRESPA hierarchical timestepping and SGI FFT support were implemented. The rRESPA work was also important for RuBisCO modeling.

Benchmarking on the ASCI Red, Ross, Cheetah, and Liberty machines was performend.

6.7 Q2 Progress and Dynamics

The following tasks were worked on in Q2 (Jan-Mar) of FY05 as listed in the goals outlined above.

A 2nd LAMMPS release was made on 17 Jan 2005.

Algorithmic improvements: Rigid body dynamics was improved, and the targeted molecular dynamics were enhanced.

Speed enhancements: The task for pre-tabulated Coulombic forces for long-range solvers was implemented. This offered a significant CPU improvement for protein simulations such as the RuBisCO model discussed above.

New featues: The task for computation and dumping of per-atom energy and stress was implemented. This is important for the metal-on-silica modeling.

Benchmarking on the HPCx machine was performeed.

6.8 Q3 Progress and Dynamics

The following tasks were worked on in Q3 (Apr-Jun) of FY05 as listed in the goals outlined above.

A 3rd LAMMPS release was made on 3 Jun 2005.

New features: The tasks for hybrid pair and bond potentials, pre-tabulated pairwise potentials, and class 2 inter-molecular potentials were implemented along with various other pair potentials (Yukawa, Morse, Buckingham). The hybrid potentials are critical to enabling the metal-on-silica studies described above to be performed.

Benchmarking on the Spirit and Blue Gene Light machines was performed.

6.9 Q4 Progress and Dynamics

The following tasks were worked on in Q4 (Jul-Sep) of FY05 as listed in the goals outlined above.

Speed enhancements: Benchmarking was performed for the pre-tabulated Coulombic forces for long-range solvers (implemented in Q2) and rRESPA hierarchical timestepper (implemnted in Q1), as summarized in Table 12. The 32K-atom fixed-size rhodopsin protein benchmark described above (see Figure 41) was run on the Liberty cluster at Sandia. The table option computed short-range Coulombic forces via linear interpolation from a 4096-length table indexed by a fast bitmapping technique. The rRESPA option used a 3-level hierarchical timestep scheme with a fast timestep of 1.0 fmsec (acceptable with SHAKE) and a slow timestep of 4.0 fmsec (where long-range forces are evaluated).

Processors	no Table, no rRESPA	Table, no rRESPA	no Table, rRESPA	Table, rRESPA
1	247.3	166.7	170.7	137.8
8	30.0	21.6	22.1	17.1
64	4.81	3.52	3.81	3.09

Table 12: CPU time for 100 timesteps of the rhodopsin protein benchmark on the Liberty machine with short-range Coulombic tabling and rRESPA hierarchical timestepping options turned off and on.

The effect of these two options on the overall run-time (which includes other computations) is shown in the table. The one-processor timing decreased from 247 to 138 seconds (a 44% improvement). The 64-processor timing decreased

from 4.8 to 3.1 seconds (a 36% improvement). These improvements are being exploited by the RuBisCO protein modeling described above to enable longer, more comprehensive simulations.

New features: Upgrades to the current 3 Jun 2005 version of LAMMPS were made to improve the NVT and NPT integrators. An NPH integrator was also added. These improvements enable better simulation of systems at constant temperature and/or pressure.

The task for point dipole force field development is nearly complete. The short-range part (both with a cutoff and for coupling to a long-range solver) is fully implemented and tested. The long-range dipole solver has been implemented as a modification to the PPPM algorithm in LAMMPS, where each dipole is treated as two (slightly) separated point-charges. We are still investigating the accuracy of this approximation and hope that the new options (force fields, long-range solver, dipole atom style, modified integrator and thermodynamic routines) will be released shortly as part of a new LAMMPS version.

The task for implementation of a Python-based pre- and post-processor for LAMMPS was completed. The Pizza.py package [34] was released for public download on 7 Sept 2005, with documentation, example scripts, screehshots, and movies all available on-line. The tools in the Pizza.py toolkit enable (a) LAMMPS output to be visualized and animated via Raster3d, RasMol or SVG files, (b) LAMMPS log-file thermodynamic data to be plotted and anlyzed via GnuPlot or MatLab, and (c) a variety of other LAMMPS related tasks to be performed interactively or scripted for automated processing. The toolkit is designed to enable its easy use with other simulation packages as well.

7 S3D SciDAC

7.1 Objectives

The objective of this document is to provide an overview and roadmap to the performance of benchmark test simulations as the 2005 Joule Software Effectiveness Study. The software to be used in this work is S3D, a high-fidelity finite-difference solver for compressible reacting flows with detailed chemistry. The overall architecture of the code is shared by the INCITE project group led by Jacqueline H. Chen at Sandia National Laboratories; however, the primary tasks undertaken by this study are distinguished from those by the INCITE project in that the additional physical sub-models, such as the radiative heat transfer and spray dynamics models, are tested for their scientific fidelity and computational efficiencies in parallel computing platform for a wide range of problem scales. The radiation models will provide useful insights into the performance of two comparable gray-gas approaches: the discrete ordinate (DOM) and discrete transfer (DTM) methods. The spray model will serve as a benchmark test case to identify issues related to the coupling between Lagrangian particle-tracking method and the Eulerian flow solver. The benchmark simulations under two different DOE computing platforms will allow a direct assessment of the exe-

cution efficiencies of the code under different hardware architectures at various scales. Simultaneously, new scientific findings will be investigated and analyzed based the temporally- and spatially-resolved data fields including detailed information of thermo-chemical variables. These goals will be accomplished by clearly identifying the metrics and observables based on computer science and physical science perspectives.

7.2 Code Description

S3D [109] is a parallel F90-based code, originally developed at Sandia National Laboratories with Basic Energy Sciences sponsorship, that has evolved through the collaboration between Sandia National Laboratories, University of Michigan, University of Maryland, University of Wisconsin, and Pittsburgh Supercomputing Center. The code solves multi-dimensional compressible reacting flow equations by incorporating high-order finite difference discretization [110, 111], high-order Runge-Kutta explicit/implicit time integration [112,113], and characteristic boundary conditions treatment [114, 115], in a structured uniform/non-uniform computational mesh system. Detailed reacting species equations are solved with elementary gas-phase chemical kinetics and multi-component molecular transport based on the approach used in the standard CHEMKIN package [116], but has been restructured to the F90 format optimized for better performance. The entire code has a modular structure to facilitate individual submodule development and subsequent code integration. The code is entirely MPI-compatible and the scalability of the basic solver has been tested under various hardware platforms [109].

The modular F90 code was then further modified to fit into an advanced software framework, known as the grid adaptive computational engine (GrACE) [117, 118]. GrACE is an MPP framework targeted for AMR applications and includes load-balancing capabilities. Thus far, a library of F90 wrappers for GrACE was developed such that GrACE can be called from within the F90 programs. This library allows a maximum use of the S3D subroutines written in F90.

Recent developments in the physical submodels are summarized as follows:

- Three radiation models based on the spectrally-averaged gray gas approximation have been implemented: First, the optically thin gas approximation leads to a simple evaluation of the radiative heat source term in terms of the Planck mean absorption coefficient prescribed as a function of mixture composition and temperature. At an advanced level, the discrete ordinate method (DOM) [119], which has an advantage of low cost and ease of integration into the finite-difference grid structure, has been fully adopted into S3D with MPI compatibility. In this approach, the radiative transfer equations (RTE) are discretized into several ordinate directions, and the integral contribution from each ordinate is approximated by a Gaussian quadrature summation. This method is also referred to as an Sn approximation, where n represents the order related to the number

of discrete ordinates. As an alternative approach, the discrete transfer method (DTM) [120] has also been developed. DTM is a ray-tracing method where the RTE is integrated along a set of representative rays. The adopted DTM formulation neglects ray scattering and the grid-based radiation power density is reconstructed from the ray-based decomposition using a simplified projection operator. Both of these models are coupled with a soot formation model based on a two-variable formulation for soot mass fraction and soot number density [121, 122]. The source terms in these equations account for basic physical and chemical soot processes including soot nucleation, surface mass growth, particle coagulation, and surface oxidation by O2/OH species.

- To understand the mechanism of spray combustion and its dependency on droplet evaporation, turbulence mixing and ignition, a spray module has also been developed. For dilute phase of spray, a Lagrangian method is employed to track the individual droplets, and is embedded into the Eulerian framework for the gas-phase flow in S3D. This PICell (Particle-In-Cell) method is widely used in the field of spray/particle fluid dynamics [123]. A fourth-order linear interpolation scheme is adopted to identify the local gas properties at the droplet location [124]. A general method is adopted to distribute the source terms according to an arbitrarily defined basis function, such that the distribution of drop source terms is smooth and independent of the grid size [125].

7.3 Overview of the TSTC Project

The long-term goal of the SciDAC TSTC project for FY04-07 is to extend the S3D code with new physical and algorithmic modules. The ultimate demonstration of the task will be simulations of partially premixed turbulent flames with liquid spray description, thereby achieving the full potential of the state-of-the-art DNS capability as a companion of detailed experimental studies. The specific objectives of this proposal include:

- To develop and complete the high-fidelity numerical algorithms under the component-based and parallel computing platform. This includes high-order, implicit/explicit (IMEX) stiff time integrators based on additive Runge-Kutta, and the immersed boundary method (IBM) for solid body representation associated with high-order interpolation schemes.

- To expand and upgrade the physical submodels to describe the underlying mechanisms with great details. The existing modules of radiation, soot, and spray evaporation model will be further enhanced to allow direct comparisons against experimental studies. In particular, the current two-equation semi-empirical soot model will be modified to incorporate detailed soot formation kinetics using the method of moments. Improved spray models will add more realism to represent direct injection of spray and droplet distortion effects.

- The grid module in S3D has been re-designed to be compatible with the GrACE framework [117, 118]. In collaboration with another SciDAC project, CFRFS [126], S3D will incorporate the adaptive mesh refinement (AMR) capability.

- To demonstrate the capability of the terascale DNS code in investigating fundamental science issues by several pilot simulations of canonical flames observed in turbulent combustion. The pilot configurations proposed for TSTC Phase II include partially-premixed turbulent counterflow and jet flames, and turbulent spray jet evaporation and ignition problems. The S3D DNS code will further allow access to various post-processing functionalities for effective data-mining and visualization that are being developed under BES Chemical Sciences core program .

7.4 Science Case: Ethylene-Air Counterflow Nonpremixed Flames with Soot and Radiation

Based on the current phase of the code development, the main scope of the core science case study for FY05 is the benchmark testing of various physical sub-modules described in 7.2 under several DOE computing hardware platforms for a wide range of problem scales. The details of the scope, observables, and deliverables of each task are described here and in the following case study.

Predictive capability for the pollutant formation processes in combustion systems will continue to be of strong interest in DOEs mission considering energy utilization and environmental concerns. While the chemical kinetics for the NOx formation in hydrocarbon fuels are considered reasonably well established, the detailed description of the soot formation process is far from being mature. Moreover, most sooting flames are highly radiative involving optically thick medium, hence it is also very important to capture the essential coupling between the soot chemistry and radiative heat transfer in order to describe the fundamental physico-chemical processes. The proposed case study will provide valuable information regarding the level of fidelity presented by different models, their computational efficiency under the S3D code architecture and MPI platform, and eventually understanding of the fundamental mechanisms of soot production in turbulent nonpremixed combustion.

The conservation equations for mass, momentum, energy, and species equations solved in S3D are written in the compressible flow formulation as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{64}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re_a}\frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial u_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)\right) \tag{65}$$

$$\frac{\partial}{\partial t}(e_t) + \frac{\partial}{\partial x_j}[(e_t+p)u_j] = \frac{1}{Re_a Pr}\frac{\partial}{\partial x_j}(u_i\tau_{ij}) \tag{66}$$
$$+ \frac{1}{Re_a Sc_i}\sum_{k=1}^{N-1}\left((h_i - h_{N_2})\frac{\partial}{\partial x_j}\left(\mu\frac{\partial Y_k}{\partial x_j}\right) + (c_{p,j} - c_{p,N_2})\mu\frac{\partial Y_k}{\partial x_j}\frac{\partial T}{\partial x_j}\right) - \nabla\cdot\mathbf{q}_{rad}$$

$$\frac{\partial}{\partial t}(\rho Y_k) + \frac{\partial}{\partial x_j}(\rho u_j Y_k) = \frac{1}{Re_a Sc_i}\frac{\partial}{\partial x_j}\left(\mu\frac{\partial Y_k}{\partial x_j}\right) + \omega_k \tag{67}$$

where the total energy is defined as

$$e_t = \rho\frac{u^2+\nu^2+w^2}{2} + \rho\sum_{k=1}^{N-1} h_k Y_k - p. \tag{68}$$

The last term in (66), $-\nabla\cdot\mathbf{q}_{rad}$, is the radiative heat transfer term which needs to be modeled. If all the gases are assumed gray (the absorption coefficients are independent of the wavelength), this term is expressed as:

$$-\nabla\cdot\mathbf{q}_{rad} = \kappa[\int_{4\pi} I(r,S)dS - 4\pi I_b] \tag{69}$$

where κ is the absorption coefficient, $I(r,S)$ is the directional radiation intensity at a distance r and solid angle S, and I_b is the blackbody radiation intensity.

In the present study, three radiation models are tested for their relative accuracy and computational efficiency. First, the optically thin radiation model (OTM) is one of the simplest methods for calculating radiation. If the gases are optically thin, the integral term in (69) vanishes such that the radiative heat transfer term is simplified to an explicit expression:

$$-\nabla\cdot\mathbf{q}_{rad} = -4\sigma\kappa(T^4 - T_\infty^4) \tag{70}$$

This model will serve as a baseline study case, while its accuracy will degrade as the gas medium in the reacting flows becomes more radiatively active. If the gases are optically thick, a more comprehensive treatment of radiative transfer equations (RTE) is needed to determine the radiation intensity. Two alternative approaches are considered here.

The discrete ordinate method (DOM) was first proposed by Chandrasekhar to solve stellar and atmospheric radiation. The RTEs for DOM is represented by a set of equations for an intensity that is angularly averaged over each of a finite number of ordinate directions. Intergrals over a range of solid angles are approximated by a weighted sum of the angular quantities. Thus, the discrete equations of transfer are give by

$$\alpha_m\frac{\partial i_m}{\partial x} + \delta_m\frac{\partial i_m}{\partial y} + \gamma_m\frac{\partial i_m}{\partial z} = a(S)i_b(S) - (a(S)+\sigma_s(S))i_m(S) \tag{71}$$
$$+ \frac{\sigma_s(S)}{4\pi}\sum_{m'} w_{m'} i_{m'}(S)\phi_{m'm}(S)$$

By solving the finite number of equations for the intensity by iteration, the radiation intensity and heat loss at every location and time can be found.

Alternatively, the discrete transfer method (DTM) is composed of determining the intensity for each of N rays arriving at each surface element in an enclosure. The rays and their associated solid angles are equally distributed over the surface of a hemisphere centered over the receiving element. The point of origin of each ray is determined by finding the intersection of the ray with the enclosure boundary. The source function, temperature, and properties are usually assumed to be constant within a given volume element. The radiation transfer equation along each ray can be placed in the form of a recurrence relation,

$$i_{n+1} = i_n e^{-\kappa S_n} + I_n(1 - e^{-\kappa S_n}) \tag{72}$$

where i_{n+1} is the ray intensity leaving boundary $n+1$ of a volume element and is equal to the ray intensity i_n crossing boundary n and attenuated along the path S_n within the element, plus the increase from the source function I_n within the element.

In the present simulations, the soot formation processes are described using a semi-empirical model developed by Moss et al. [121, 122]. In addition to the above conservation equations, the transport equations of soot number density and soot mass fraction are solved, which are given by:

$$\frac{\partial}{\partial t}(\frac{n}{N_0}) + \frac{\partial}{\partial x_i}(\frac{n}{N_0}u_i) = \frac{\partial}{\partial x_i}[\frac{\nu}{Sc}\frac{\partial(n/N_0)}{\partial x_i}] - \frac{\partial}{\partial x_i}(\frac{n}{N_0}V_{t,i}) + \dot{\omega}_{n/N_0} \tag{73}$$

$$\frac{\partial(\rho Y_s)}{\partial t} + \frac{\partial(\rho Y_s u_i)}{\partial x_i} = \frac{\partial}{\partial x_i}[\rho\frac{\nu}{Sc}\frac{\partial Y_s}{\partial x_i}] - \frac{\partial(\rho Y_s V_{t,i})}{\partial x_i} + \dot{\omega}_{\rho Y_s} \tag{74}$$

where the soot formulation rates are given by:

$$\begin{aligned}
&\dot{\omega}_{n/N_0} = c_\alpha \rho^2 T^{1/2} X_c exp(-T_\alpha/T) - c_\beta T^{1/2}(n/N_0)^2 \qquad (75)\\
&\dot{\omega}_{\rho Y_s} = c_\gamma \rho T^{1/2} X_c exp(-T_\gamma/T)n + c_\delta c_\alpha \rho^2 T^{1/2} X_c exp(-T_\alpha/T) - \dot{\omega}_{0\chi} S\\
&\dot{\omega}_{0\chi} = 1.085 \times 10^5 X_{O_2} T^{-1/2} exp(19778/T).
\end{aligned}$$

7.4.1 Case Study

In this test, three radiation models developed under the project and their coupling with soot formation model are examined for the performance and scalability. The optically thin radiation model (OTM) is a simple approach to provide a baseline case. DTM and DOM need a significant amount of calculation time for the iterative solution process, so that it is important to test the parallelized version of the code and assess their performance at different problem configurations and sizes. The basic test configuration is a two-dimensional steady nonpremixed flame subjected to a transient interaction with two pairs

of counter-rotating vortices. One-step ethylene-air reaction mechanism is used and the two-variable soot formation model is implemented to demonstrate the coupled effects of soot and radiation.

7.4.2 Parameters

- Number of processors: 1, 4, 16, 64, and 128
- Radiation models: DOM and DTM
- Computing hardware: Cheetah (CCS) and Seaborg (NERSC)

7.4.3 Results

Two pairs of counter-rotating vortices are superimposed on the steady counterflow nonpremixed ethylene-air flame. The test domain size is $2.48cm \times 2.48cm$ with 400×400 grid resolution 7.4.4 for detailed description). For the scalability test, calculations are done up to only 1000 time steps to compare two different radiation models. The time steps are fixed constant at $\delta t = 10^{-7} sec$. The radiation field is not calculated in every Runge-Kutta stage, but only once in every time step.

Fig[44] shows the scalability of DOM and DTM on two different platforms. The speed-up is with respect to the CPU time for DTM with a single processor on Seaborg (the slowest one), showing the relative speeds between different models and different processors more clearly. For reference, the ideal linear speed-up lines are drawn based on DOM on Cheetah (fastest) and DTM on Seaborg (slowest). It is seen that the scalability is significantly reduced in both models. This is because the two advanced radiation models require a large number of iterations and MPI communications for the radiation field to converge. Comparing the two radiation models, it is found that DOM is somewhat faster than DTM by factor of 1.13 to 2.42 depending on the platform and number of processors. In terms of the relative speed on two platforms, Cheetah (based on IBM Power 4) is in general faster than Seaborg (based on IBM Power 3) by factor of 1.37 to 2.32 depending on the radiation model and number of processors. For DTM, the scalability on Cheetah is poorer than that on Seaborg; For DOM, the scalability on two platforms is comparable.

The profile of individual code modules on their contribution to the total CPU time has been analyzed using TAU (Tuning and Analysis Utilities) [127]. Fig[45] shows the inclusive percentage of CPU time consumption of each subroutine for DOM with 64 processors on Seaborg. Fig[46] shows the result for DTM under the same test conditions. Comparing the two radiation models, DOM (denoted by DOM PHASE) used about 51

7.4.4 Flame-vortex interaction

One of the science goals of the present study is to understand the fundamental characteristics of interaction between turbulence and chemistry in the presence

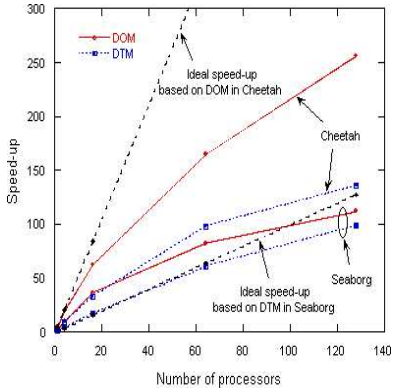

Figure 44: The speed-up of DOM and DTM on two platforms. All quantities are normalized by the CPU time for DTM with a single processor on Seaborg.

Figure 45: Inclusive percentage of CPU time consumption of each subroutine averaged over all processors using DOM with 64 processors on Seaborg.

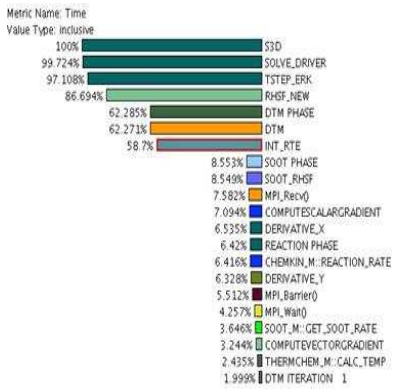

Figure 46: Inclusive percentage of CPU time consumption of each subroutine averaged over all processors using DTM with 64 processors on Seaborg.

of soot and radiation. As a preliminary model problem, we first study the interaction of counter-rotating vortices and laminar nonpremixed flame, which represents a canonical configuration to study more complex turbulent combustion phenomena. To avoid excessive complication of the problem, a one-step ethylene-air chemistry with the semi-empirical soot model shown in (73 - 75) was used. The radiation model was also simplified to OTM.

The left-most images in Fig[47,48] show the initial condition of the simulation. A steady laminar diffusion flame is established in an opposed flow (inflow at the left and right boundaries), and two pairs of counter-rotating vortices (shown in black line contours) are superimposed. The vortex pairs are subsequently convected onto the flame by the mean flow as well as self-induced drift, thereby penetrating through the flame with vigorous interaction until the vortices become dissipated away by viscous diffusion.

Fig[47] and Fig[48] show the temporal variations of (a) temperature and vorticity and (b) soot volume fraction at t = 0, 3, 5, and 20 msec, respectively. As discussed earlier, a grid resolution of 400×400 was used and the time step was set at $\delta t = 10^{-7}$sec. The simulation was run 200,000 time steps, which takes approximately 30 hours with 64 processors on Seaborg. As the vortices are convected to the flame, the soot quenching occurs at the center part of the flame (at t = 3 msec) even the flame is not completely extinguished. As vortices pass through the boundaries, the highly stretched flame restores the original nonpremixed flame and soot layer also recover the original shape of the steady solution.

Fig[49] shows temporal variations of the local maximum soot volume fraction, f_ν, and soot number density, N_{soot}. Comparing Fig[48] and Fig[49], it is recognized that the maximum N_{soot} depends strongly on the maximum temperature. The instantaneous burst in peak temperature zones toward the fuel side of the flame results in a rise in the production of soot particles (at t = 3

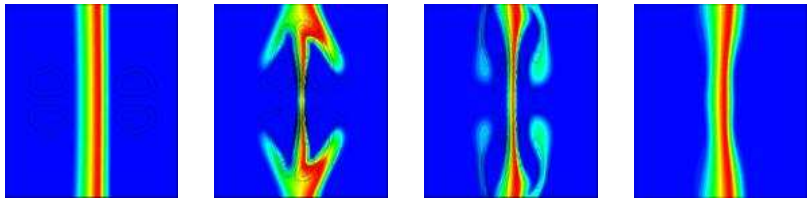

Figure 47: temperature (flood) and vorticity (black lines) - left to right, t = 0, 3, 5, and 20 msec.

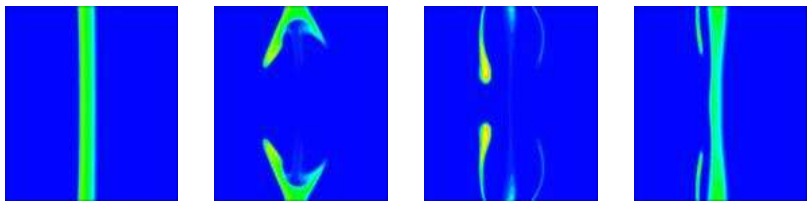

Figure 48: soot volume fraction, f_ν, with the optically thin radiation model -left to right, t = 0, 3, 5, and 20 msec.

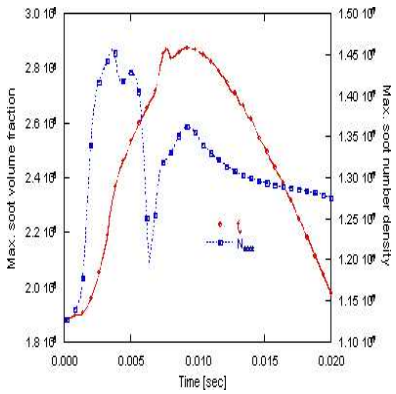

Figure 49: Temporal variations of the local maximum soot volume fraction, f_ν and soot number density, N_{soot}

msec). On the other hand, the maximum soot volume fraction does not follow the number density but is rather closely related to the soot oxidation process. In other words, the maximum f_ν appears in low temperature region where the soot oxidation process is turned off at t = 5 msec. After the vortices effectively carry the soot particles out of a high temperature region into a low temperature region, the soot oxidation rate is reduced and there is sufficient residence time to increase the soot volume fraction. Consequently, the maximum soot volume fraction is observed in the low temperature region as seen in Fig[48] at t = 5 msec.

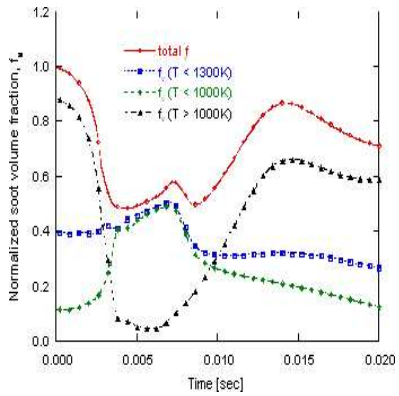

Figure 50: Temporal variations of the volume-integrated soot volume fraction.

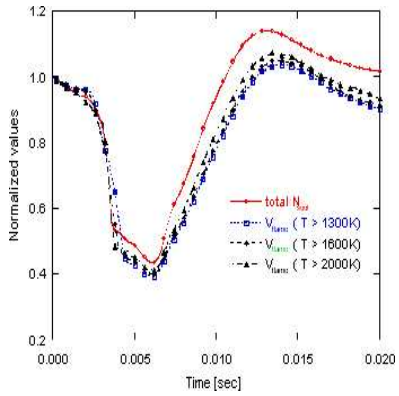

Figure 51: Temporal variations of the integrated soot number density and flame volume.

Consistent behavior is also found in the volume-integrated soot volume fraction. Fig[50] shows temporal variations of the volume-integrated f_ν in different temperature regions. For example, $f_\nu(T < 1000K)$ represents the total soot

volume fraction in the region at which temperature is less than $1000K$. Even if soot extinction occurs during t = 3 and 5 msec in Fig[48], $f_\nu(T < 1000K)$ is increased during that period, which implies that soot is pushed toward the low temperature region by vortices and grows into a larger volume.

Finally, Fig[51] shows the temporal variations of the flame volume. The flame volume is defined such that $V_{flame}(T > 2000K)$ represents the total volume within the domain that are at temperatures above $2000K$. Also overlaid is the volume-integrated soot number density (red). The results clearly show that the total soot number density correlates strongly on the volume of high temperature region, primarily due to the fact that the maximum soot number density occurs at the maximum temperature.

7.4.5 Summary of counterflow nonpremixed flame study

During this period of the project, we have successfully developed and integrated three radiative transfer models and a semi-empirical soot model into the high-fidelity direct simulation code, S3D, in order to study the fundamental issues of soot-radiation-turbulence interactions occurring in nonpremixed combustion. The new physical models allow representation of sooting/radiating nonpremixed flames with substantially improved realism. As demonstrated in 7.4.4, the detailed simulation data provides accurate information about the local and instantaneous soot behavior and how it is affected by the neighboring conditions. The generated numerical databases will prove valuable in developing more physically-based soot formation submodels in larger scale device simulations using RANS and LES.

Computational performance comparison was made primarily between the two sophisticated radiation models. It is found that DOM performs somewhat better than DTM for a given overall accuracy. Both methods require a number of iterations and MPI communications, thereby adversely affecting the scalability with a large number of processors. It is noted, however, that the test problem size was not large enough compared to the ultimate production run cases that will be done in the future. As a result, the actual load per processor for the 64-processor case was unrealistically low. Therefore, it is expected that the scalability will improve as the problem size increases. The overall computational overhead of the radiation models ranges within 50-60% in addition to the main code, hence it is expected to be applicable to large scale production runs.

Future work will include further refinement of the existing physical models for better MPI performance, and the development of high-fidelity soot model based on detailed chemical processes.

7.5 Science Case: Fuel Spray Injection and Mixing

Fuel spray injection is a very important phenomenon in the internal combustion engine application. The evaporation rate of fuel spray and the mixing between the fuel vapor and the carrier gas dominate many processes, such as the ignition timing, the production of particulate matters and the emissions of NOx and

soot. For engineering applications, the process of fuel spray injection is usually modeled using the Reynolds-averaged Navier-Stokes (RANS) or large eddy simulation (LES) approaches with limited resolution of fine-scale physics. To fully understand the fundamental mechanisms associated with the sub-processes of evaporation, mixing, ignition and combustion, it is essential to resolve all relevant temporal and spatial scales with minimal numerical dissipation. Therefore, the objective of this study is to develop high-fidelity modeling to describe turbulence and spray dynamics, thereby investigating the effects of injected fuel spray on carrier gas flow field. The simulation results will provide better understanding of the transport mechanisms of mass, momentum and energy between the carrier gas and the fuel spray. The high resolution data will also serve as valuable resources to develop and validate new LES or RANS models for fuel injection flows.

7.5.1 Transport Equations

Fuel spray droplets are treated in a Lagrangian manner and assumed to be spherical, with a constant density. The drag force is determined by the Stokesian drag law and the evaporation is calculated according to the lumped heat capacitance approximation inside the fuel droplet. The governing equations for the droplet of radius, r_d, velocity, u_d, mass, m_d, and temperature, T_d, in non-dimensional form are:

$$\frac{dx_{d,i}}{dt} = u_{d,i} \tag{76}$$

$$m_d \frac{du_{d,i}}{dt} = F_{gd,i} \tag{77}$$

$$\frac{dm_d}{dt} = -2\pi\rho D r_d Sh_d B_M \tag{78}$$

$$\frac{dT_d}{dt} = \frac{4\pi r_d^2}{m_d c_{liq}}[\rho_d \frac{dr_d}{dt} L_{vap} + \frac{\lambda_g (T - T_d) Nu_d}{2r_d}] \tag{79}$$

where $F_{gd,i}$ is the drag force by the carrier fluid, ρD is the diffusivity of fuel droplets in the carrier fluid, Sh_d is the Sherwood number, B_M is the mass transfer number, c_{liq} is the specific heat, L_{vap} is the latent heat of fuel droplets, λg is the thermal conductivity, and Nu_d is the Nusselt number.

The carrier fluid is considered to be Newtonian and compressible. It contains N species of gases and the mixture satisfies the equation of state for an ideal gas. The molecular weight, mass fraction and the specific enthalpy of each species are denoted by W_i, Y_i and h_i, respectively. The instantaneous pressure,

temperature, density and velocity are expressed by p, T, ρ and u_i. The total energy is defined as

$$e_t = \rho\frac{u^2+\nu^2+w^2}{2} + \rho\sum_{i=1}^{N}(h_iY_i) - p \tag{80}$$

The ideal gas equation of state is

$$p = \rho RT \tag{81}$$

The transport equations for mass, momentum, energy and species can be written as

$$\frac{\partial\rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = \psi_\rho \tag{82}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re_a}\frac{\partial\tau_{ij}}{\partial x_j} + \psi_{u_i} \tag{83}$$

where

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) \tag{84}$$

$$\begin{aligned} & \frac{\partial e_t}{\partial t} + \frac{\partial}{\partial x_j}[(e_t+p)u_j] = \psi_e \\ + \quad & \frac{1}{Re_aPr}\frac{\partial}{\partial x_j}(\mu c_p\frac{\partial T}{\partial x_j}) + \frac{1}{Re_aPr}\frac{\partial}{\partial x_j}(u_i\tau_{ij}) \\ + \quad & \frac{1}{Re_aSc_i}\sum_{i=1}^{N-1}[(h_i-h_{N_2})\frac{\partial}{\partial x_j}(\mu\frac{\partial Y_i}{\partial x_j}) + (c_{p,i}-c_{p,N_2})\mu\frac{\partial Y_i}{\partial x_j}\frac{\partial T}{\partial x_j}] \end{aligned} \tag{85}$$

$$\frac{\partial}{\partial t}(\rho Y_i) + \frac{\partial}{\partial x_j}(\rho u_j Y_i) = \frac{1}{Re_aSc_i}\frac{\partial}{\partial x_j}(\mu\frac{\partial Y_i}{\partial x_j}) + \omega_i + \psi_\rho\delta_{i,vapor} \tag{86}$$

In above equations, ψ_ρ, ψ_{u_i}, and ψ_e are the source terms contributed by the fuel droplets for the exchange of mass, momentum and energy, respectively, which are expressed as

$$\psi_\rho = -\frac{1}{\Delta V}\sum_{k=1}^{n_d}(\frac{dm_d}{dt})_k \tag{87}$$

$$\psi_{u_i} = -\frac{1}{\Delta V}\sum_{k=1}^{n_d}(F_{gd,i} + \frac{dm_d}{dt}u_{d,i})_k \tag{88}$$

$$\psi_e = -\frac{1}{\Delta V}\sum_{k=1}^{n_d}(\frac{d(m_d h_d)}{dt} + F_{gd,i}u_{d,i} + \frac{dm_d}{dt}(\frac{1}{2}u_{d,i}^2))_k \tag{89}$$

where the summations occur because the source terms are appropriately distributed to the gas phase control volumes, ΔV, using local integrals. These integrals use a uniform basis function with a characteristic length scale equal to $2\Delta x$ and centered at each drop location.

The fuel spray module was developed under the support of the DOE SciDAC project in the Engine Research Center at the University of Wisconsin-Madison since 2001. The main goal to develop the liquid fuel spray module for the S3D code is to enhance the capacity of S3D to deal with soot particles and liquid fuel droplets. The spray module is based on Fortran 90 with MPI for the parallelism, consistent with other modules in S3D. It includes several submodules to calculate the evaporation rate, interpolate the gas properties at the droplet location, distribute the source terms in the governing equations for the carrier gas, and relocate droplets when they move from one processor to another processor. The current version of the module has about 14,000 lines. The source code can be found on NERSC and the CCS.

7.5.2 Case Study

We carried our simulations for the injection of fuel spray with an injection velocity of 100 m/s. The initial gas temperature is $1000K$ and the pressure is at the atmospheric condition. The reason for the choice of a high gas temperature is to increase the evaporation rate of fuel spray.

The effect of the fuel droplet size on the evolution of the carrier gas field was investigated numerically. The effectiveness of the spray module was examined on two computer platforms (Seaborg at NERSC and Cheetah at CCS) with a large number of processors, in particular when the number of fuel spray droplets reaches millions. Simultaneously, an appropriate level of grid resolution to capture the relevant spray dynamics was identified.

7.5.3 Parameters

The input parameters to be specified for the case study include:

- Computational geometry (xmin, xmax, ymin,ymax, zmin, zmax specified in s3d.in)
- Grid resolution (nx, ny, nz specified in s3d.in)
- Processor number (npx, npy, npz specified in s3d.in)
- Carrier gas properties (pressure, temperature and velocity) specified in the subroutine ini_turbulent_iso.f90.

- Fuel droplet properties (velocity, temperature, droplet size , droplet number and density etc.) specified in particle.in
- Time step size (tstep_init specified in controller.in)
- Injector size (specified in the module particle_submodel_m)

7.5.4 Code Observables

- Amount of memory and disk required
- Effect of fuel droplet numbers on the iteration time
- Scalability of the fuel spray module
- Effect of grid size on the simulations

7.5.5 Physical Observables

- Evolution history of the carrier gas temperature, velocity
- Effects of parameters on spray dispersion and evaporation
- Effects of parameters on fuel-oxidizer mixing

7.5.6 Results

By changing the droplet number, the CPU usage on Seaborg is examined. The grid size is 128^3 and the droplets are distributed over the computational domain uniformly using a random function. The droplet number was varied from 0 to 10^6. This test is very important since in some applications the number of fuel droplets can reach millions. It is hoped that the CPU time required by the spray module with millions of fuel droplets is amenable to simulations on the existing computing hardware. The code was set to run for 200 steps each time and 64 processors were employed. TAU (Tuning and Analysis Utilities) [127] was used to trace the CPU usage and the performance of each subroutine. We found that, with one million fuel droplets and 128^3 grid points, 8.5 GBytes memory in total is required, while the hard disk storage requirement is about 100 GBytes. Since most nodes on Seaborg have more than 32 gbytes memory, the memory is not an issue in the case study.

The CPU usage is shown in Table[13], which shows that even there are no droplets, the spray module still uses about 2.1 percent of the total CPU time. This is due to the initialization of the spray module and some output of the statistical properties in the spray subroutines. With one million droplets, the CPU time by the spray module is 45.4 percent of the total S3D CPU time. Therefore, it is concluded that simulations with a million droplets do not impose an excessive overhead to the overall computing time. Fig[52] further shows that the CPU time taken by the spray module increases linearly with the droplet

droplet count	0	10^3	10^4	10^5	10^6
CPU time %	2.1	11.2	17.5	21.7	45.4

Table 13: CPU time used by the spray module as a percentage to the total CPU time of S3D

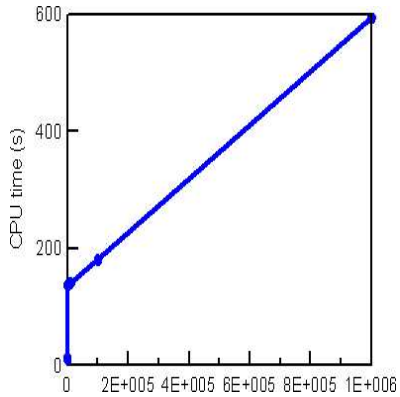

Figure 52: Effect of droplet number on the CPU time of spray module with 64 processors

number, except for a extremely small number of droplets when the basic CPU time overhead (initialization and output) becomes dominant.

The performance of the individual code modules are analyzed on two different computer platforms, Cheetah and Seaborg. A spray injection test case was set for a domain size of $8cm \times 4cm \times 4cm$ in the x, y, and z-direction, for which a grid size of $256 \times 128 \times 128$ is used. A total of 64 processors were used on both Seaborg and Cheetah. The total computation time was approximately 10 hours on both platforms. The breakdown of the average CPU usages (inclusive) for individual modules is produced by TAU, as shown in Figs[53,54]. The spray module (particle_driver which is called from rhsf_new) is found to be slightly more efficient on Seaborg than on Cheetah by approximately 4%. It is also recognized that approximately 17-19% of the CPU time is used by the MPI communication (MPI_Recv and MPI_Wait), which may need further investigation for improvement.

To investigate the effect of grid size, for a 2D domain size of $8cm \times 4cm$, three levels of grids (256×128, 512×256, and 1024×512) were used in the calculations. The initial gas temperature is $1000K$ and the pressure is at 1 atm. The mean flow is 5 m/s in the x-direction through the injector with a size of 0.5 cm. N-heptane was chosen as the fuel, with the mass flow rate of 1.0 g/s. The fuel droplet radius is 10 mm and its initial temperature is at 300 K. Fuel spray is injected from the left side of the domain with an injection velocity of 100 m/s. The velocity contours for the slected solution fields at t = 2.46 ms

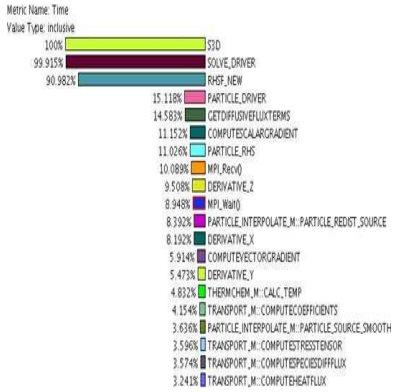

Figure 53: CPU time usage on Cheetah with 64 processors

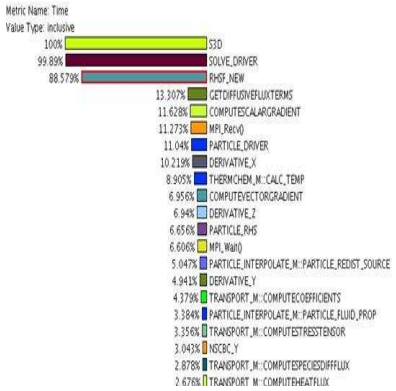

Figure 54: CPU time usage on Seaborg with 64 processors

are show in Fig[55]. With 256×128 grid points, the results show some spurious waves emanating from the tip of the spray due to the lack of spatial resolution.

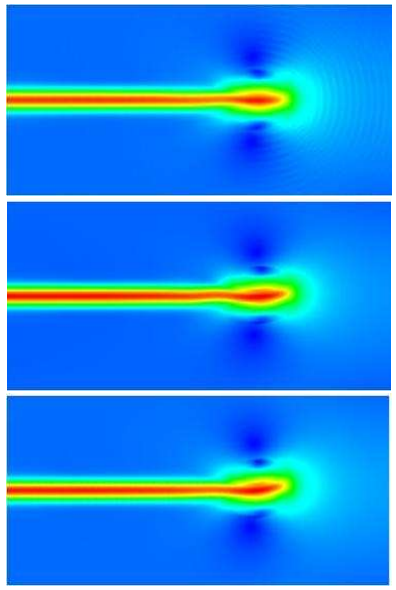

Figure 55: Velocity contours at t = 2.46 ms for three grids: 256×128(top),512×256(middle), 1024×512(bottom)

Fuel spray distribution and the mixture fraction contour for the case with 1024×512 grid points are shown in Fig[56].

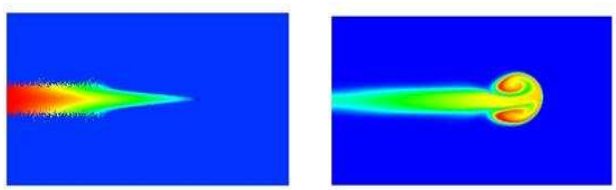

Figure 56: Fuel spray distribution (left) and mixture fraction contour (right) at t=2.46 ms for the case with 1024×512 grid points

To investigate the effect of fuel droplet size, two simulations were carried out with 512×128 grid points. The domain size is $4cm \times 1cm$. Therefore, the grid size is the same as in the above case with 1024×512 grid points. The injector size is 0.125 cm. The mass flow rate of the fuel spray is 0.25 g/s. Other initial conditions for both the carrier gas and the fuel are the same as in the above calculations with different grid sizes. In one case, the fuel radius is 10 mm, while in the other one, the fuel radius is 5 mm. Fuel spray distributions at t=0.76 ms

are shown in Fig[57]. The plots clearly show that smaller fuel droplets (5 mm) vanish much earlier than big droplets due to evaporation.

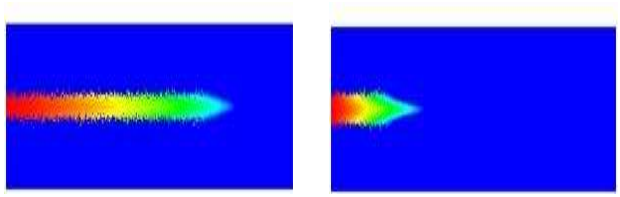

Figure 57: Fuel spray distribution at t=0.76 ms for two cases with initial droplet radiuses of 10 mm (left) and 5 mm (right).

7.5.7 Summary of fuel injection study

From the scaling runs, we found that the S3D code with the spray module is capable of large 3D simulations with millions of fuel droplets. This is a significant improvement compared to that in October 2004 (at that time, the code was not effective when more than 105 droplets were handled on a large number of processors). The improvement was achieved through the refinement of some subroutines of the spray module, such as the communication between processors and effective use of dynamic allocatable arrays. In addition, the numerical oscillation due to fuel spray droplets from the injector was also significantly reduced. We found that the numerical oscillation was caused by the discontinuous distribution of the source terms of fuel spray. This problem was alleviated by the use of a smooth dispatch function and the increase of fuel droplet number. For a production run with $512 \times 128 \times 128$ grid points on the 3D domain of $4cm \times 1cm \times 1cm$, it will take about 4 days on Cheetah or one week on Seaborg with 128 processors. Before proceeding with the production runs, however, more work still needs to be done to refine the parameters to ensure robust calculations and physically useful results.

7.6 Hardware and Software

The following hardware and software resources were used:

- NERSC: IBM SP3 Seaborg
- ORNL CCS: IBM Power4 Cheetah
- MPI
- TAU (Tuning and Analysis Utility)
- (vendor) C and Fortran compilers

7.7 Q3 Progress and Dynamics

7.7.1 Comparison of DTM and DOM in nonpremixed flames with soot formation

During Q2, we tested the performance of DTM and DOM independently with different convergence criteria, making the direct comparison for efficiency and accuracy rather inconclusive. Thus, during Q3 we tested the scalability of DTM and DOM again, with the identical convergence criterion in sooting nonpremixed flames. In this test, we checked CPU time required to obtain the steady solution for the radiative transfer equation for a given steady nonpremixed flame solution.

Since the radiative heat loss in the energy equation is the main concern for this problem, we chose the volume integrated radiative heat loss as the convergence criterion, written as: $\frac{||Q_{rad}^{n+1}-Q_{rad}^{n}||}{||Q_{rad}^{n}||} < \epsilon$ where $Q_{rad} \equiv \int -\nabla \cdot \mathbf{q_{rad}} dV$, n is the iteration number, and $\epsilon = 10^{-6}$ is chosen.

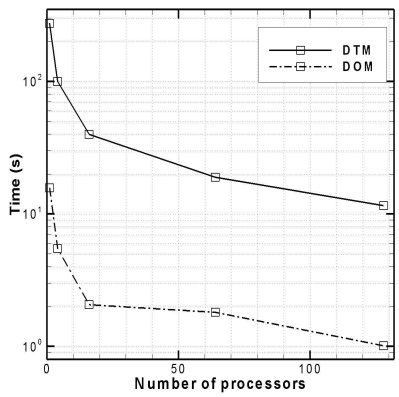

Figure 58: Net wall-clock time used on radiation solver (DTM or DOM).

Figures[58,59] show that the actual time consumed for solving the radiative transfer equation and the corresponding speed-up for DTM and DOM. In Figure[58], we can see the speed of DOM is faster than DTM by about one order of magnitude. Figure[59] shows both DTM and DOM need a larger number of iterations as the number of processors increases.

The accuracy test for DTM and DOM with the sooting nonpremixed flame was checked. Since the exact solution for this problem is not known, the result of DTM with 900 rays is chosen as the reference value. Thus the error is computed by: $\epsilon = \frac{||Q-Q_{ref}||}{||Q_{ref}||}$.

Figure[60] shows (left) the total radiative power and (right) the error relative to the reference DTM case. Comparisons were made among the OTM, DTM with different number of rays, and DOM with different S_n approximations. For this particular problem, the DTM prediction with small number of rays (32 and 64) of the total radiative loss is larger than that of OTM and DOM cases. This

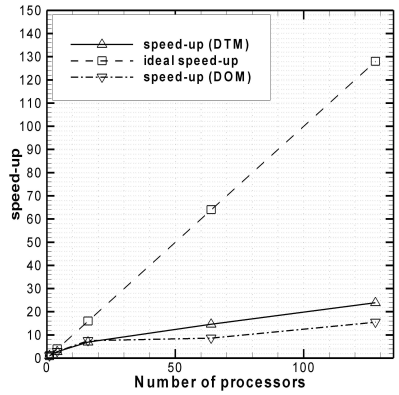

Figure 59: The speed-up of DTM and DOM with number of processors.

indicates that, when absorption is not important, the error introduced by under-resolved DTM calculations might override the absorption effect. The figure on the right confirms that DTM with 256 rays and DOM with S_4 approximation are accurate enough to predict the radiation effect with the relative error less than 1%.

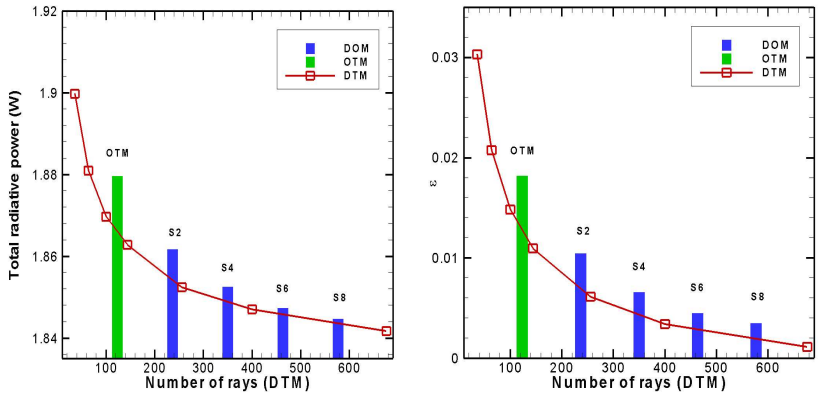

Figure 60: The total radiative power and relative error of OTM, DTM with different number of rays and DOM with different S_n approximations.

7.7.2 Effects of different formulation for the convective terms in the Navier-Stokes equations

During Q3, as an ongoing effort in the simulations of spray-turbulence interaction, we have investigated the skew-symmetric treatment for the convective terms in the Navier-Stokes equations and its effects on the solution behavior of

the spray injection simulations have been examined.

Numerical simulation of the convective terms in the Navier-Stokes equations can give rise to aliasing errors if the flow field is under-resolved. In the spectral method [[128]], the skew-symmetric treatment for the convection terms has been found to be more robust compared to the conventional conservative treatment. The convective term in the momentum equation is usually discretized in the conservative form as $\frac{\partial}{\partial x_j}(\rho u_i u_j)$.

In the skew-symmetrical form, it is discretized as [[128]]: $\frac{1}{2}\frac{\partial}{\partial x_j}(\rho u_i u_j) + \frac{1}{2}[\rho u_i \frac{\partial}{\partial x_j}(u_j) + u_j \frac{\partial}{\partial x_j}(\rho u_i)]$.

These two formulations have been tested in simulations of three-dimensional isotropic turbulent flows. As for the basic turbulence parameters, the Reynolds number based on the Taylor scale Re_λ is 62.7, the Kolgoromov length scale is 0.1 mm, and the grid size is 0.312 mm. Therefore, the solution is under-resolved. The turnover time scale is 0.0114 s. Using 64^3 grid points, calculations are carried out for both the conservative and skew-symmetric treatment for the convective terms in the Navier-Stokes equations. As shown in Figure[61], calculations using the conservative treatment for the convective terms fail after a short time due to the accumulation of the aliasing errors. On the contrary, calculations using the skew-symmetric treatment are found to sustain a much longer duration of time, showing the expected behavior of decaying turbulent kinetic energy.

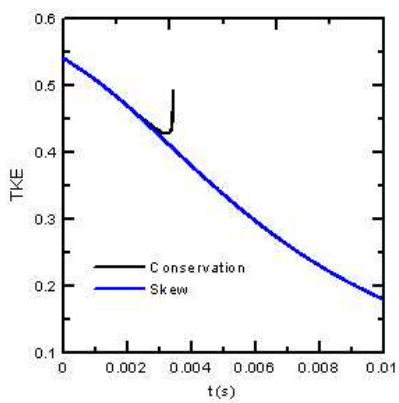

Figure 61: Evolution of the turbulence kinetic energy (TKE) with two treatments

When the computational time becomes very long, however, we found that the energy level at high wave numbers is enhanced. Since this is a decaying turbulence case, it was believed that the error originated from the aliasing errors associated with the skew-symmetric treatment. To remove aliasing errors at high wave numbers, the solution variables such as the velocity components were filtered at every 500 time steps. The turbulent kinetic energy spectrum at t=0.0055 ms is shown in Figure[62]. It is clearly demonstrated that the

dealiasing filter successfully removes the spurious accumulation of energy and thus provides more realistic turbulent kinetic energy spectrum.

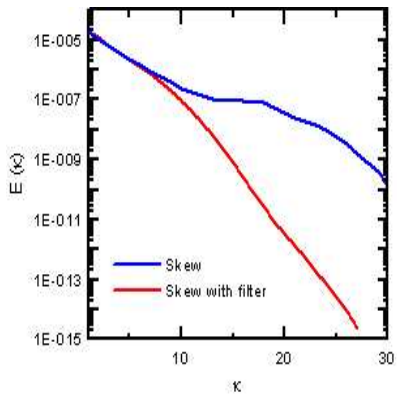

Figure 62: Turbulence energy spectra at t=0.0055 ms

It should be noted that, if filtering is used too frequently (e.g. every time step), too much energy is cut off from the system, eventually leading to a loss in the energy at lower wave numbers. To maintain numerical accuracy, we found that filtering at every 500 time steps is reasonable.

7.8 Relation to Other DOE Initiatives

The proposed study will complement the INCITE Project led by Jacqueline Chen of Sandia National Laboratories, entitled Direct Numerical Simulation of Turbulent Non-Premixed Combustion Fundamental Insights towards Predictive Modeling. The two projects share the same software components and architecture, yet differ in physical models and applications. Furthermore, the TSTC project will interface with CFRFS [126] for the development of adaptive mesh refinement.

Thanks to Professor A. Malony and Dr. S. Shende of University of Oregon for their valuable assistance in incorporating TAU into S3D.

7.9 Q4 Progress and Dynamics

In Q3, the two new radiation models, DTM and DOM, was implemented and their performance was compared. The test results showed that DOM with S4 approximation is the optimal choice in terms of solution accuracy and computational efficiency. Therefore, in Q4 the radiation model based on DOM has been applied to the flame-vortex interaction problem in order to assess the impact of the fidelity in the physical submodels on the science studies of soot dynamics in turbulent nonpremixed combustion.

7.9.1 Accuracy improvement with DOM

To assess the solution accuracy depending on the complexity of the radiation models, two different radiation models (OTM and DOM-S4) were applied and the results were compared. The flame-vortex interaction shown in Figure[**??**] was adopted as a model problem. Figure[63] shows the temporal evolution of total radiative heat loss during the interaction. While the two models had been found to yield similar predictions in the radiative heat loss under steady conditions, Figure[63] demonstrates that the discrepancy becomes significant during the rapid transition period of the strong flame-vortex interaction. In particular, during the soot extinction period (t = 3 and 5 msec), OTM is found to overestimate the radiative heat loss compared to DOM by up to a factor of two. This demonstrates the advancement in the fidelity of the science models developed during the past year.

7.9.2 Transient soot dynamics

The direct numerical simulations of flame-vortex interaction have revealed an important aspect in understanding the soot formation process occurring in turbulent flames. To investigate this issue, three different vortex strength cases were simulated, denoted as Cases A-C. Figure[64] shows the temporal variations of integrated soot volume fraction. For Case A which has the weakest vortex strength, the total volume fraction in the domain is reduced most significantly at the early stage of flame-vortex interaction. On the other hand, the strongest vortex case (Case C) yields only a small amount of reduction in the total soot volume fraction. This is interpreted from the fact that, as the vortex strength increases, more soot particles are convected from the high temperature region into the low temperature and fuel-rich region. The soot particles left in the fuel-rich regions are allowed to grow without being oxidized. Consequently, Case C shows a higher overall soot volume fraction despite the partial extinction of the flame and soot layers.

Figure[65] shows the temporal variations of integrated soot number density for Cases A-C. Comparison of Figs. 2 and 3 confirms that a substantial fraction of the increase in the soot volume fraction shown in Case C is by the surface growth in the fuel-rich region rather than by the generation of new soot particles in the flame which is turned off due to flame extinction. For Case A, even if the total volume fraction is reduced by the vortex-induced strain during the early stage of the flame-vortex interaction, it recovers the steady value after vortices pass though the domain because the weak vortices do not extinguish the flame so that the soot generation in the flame can also recover easily.

To investigate more realistic soot dynamics, we have also conducted simulations of two-dimensional counterflow turbulent diffusion flames. For this study, a domain size of $2.48cm \times 4.96cm$ was used. The inlet velocities at the two inlets were set at 1.56 m/s, yielding the overall strain rate at approximately 126s-1. DOM-S4 and the semi-empirical soot model were considered. It should also be noted that the recent development of characteristic boundary conditions [[129]]

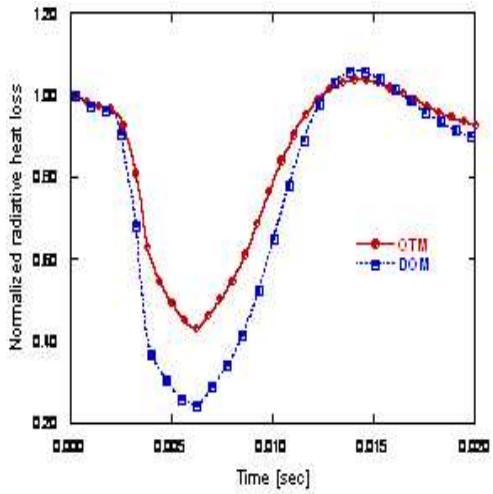

Figure 63: Temporal variations of total radiative heat loss in OTM and DOM

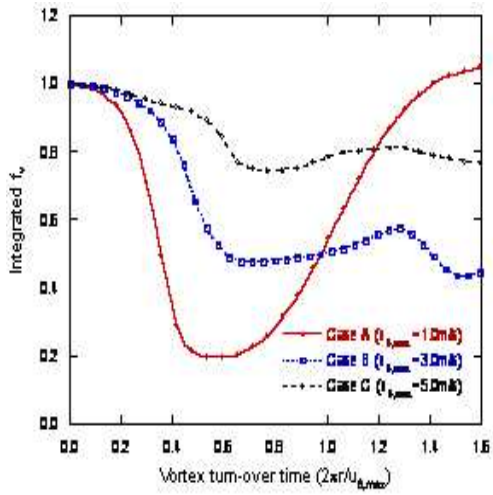

Figure 64: Temporal variations of the integrated soot volume fraction for Cases A-C.

has allowed robust and accurate simulation of turbulent counterflow flames. Figures[66,67] show the temporal variations of temperature and soot volume fraction. It is clearly seen that, as turbulence eddies fed at the inlet boundaries are convected toward the flame, the flame shape is continuously modified and the soot volume fraction is significantly affected by the neighboring flow conditions. Time evolution of the volume-integrated soot quantities is also plotted in Figure[68]. Unlike the flame-vortex interaction case, the turbulent flame yields up to 40% more soot than the steady laminar flame. This is because soot is significantly generated in low strain rate region (see soot volume fraction at t = 20 msec) even though soot is reduced in the other highly strained region.

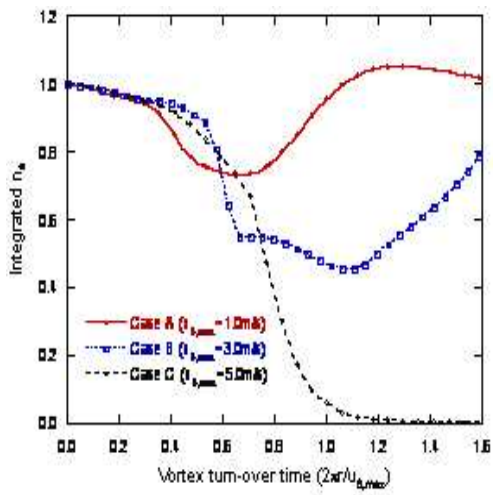

Figure 65: Temporal variations of the integrated soot number density for Cases A-C.

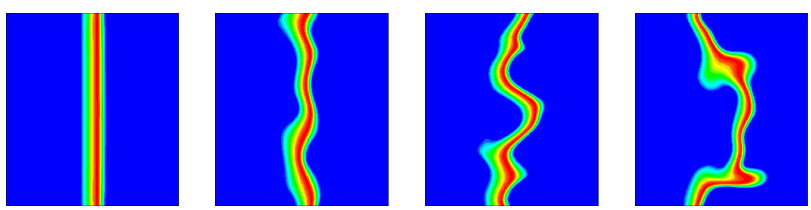

Figure 66: Temporal variations of temperature with the optically thin radiation model. From left to right, t = 0, 10, 15, and 20 msec.

The above results demonstrate that the flame-soot-radiation interaction in highly turbulent combustion can be far more complex than a simple steady description can predict. The high-fidelity simulations provide detailed information of the complex soot dynamics that provides insights toward improved submodels

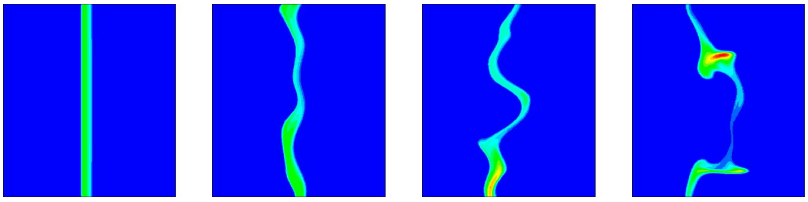

Figure 67: Temporal variations of soot volume fraction, f_{mu}, with the optically thin radiation model. From left to right, t = 0, 10, 15, and 20 msec.

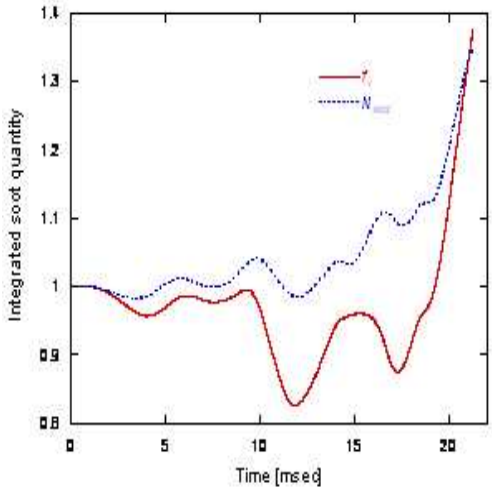

Figure 68: Temporal variations of soot number density and flame volumes

in larger-scale simulations.

7.9.3 Improved boundary conditions for spray jet simulations

The characteristic boundary conditions developed under the TSTC project [[129]] have also significantly improved the fidelity of the liquid spray jet simulation, which is a serious challenge due to the high flow speed and complex evaporation processes. In the past, we had been using the conventional hard inflow boundary conditions and the the Navier-Stokes characteristic boundary conditions (NSCBC) [130] for treating the outflow boundaries. For combustion cases, Sutherland and Kennedy [131] further improved the treatment by considering the chemical reaction source terms at the domain boundaries. Despite this improvement, the previous boundary condition treatment have revealed a significant level of spurious acoustic waves at the inflow and outflow boundaries as the spray jet reaches the outflow boundary, as shown in Figure[69].

During Q4, the new characteristic boundary conditions [129] have been tested in the spray jet simulations with the same initial flow conditions and domain size as shown in Figure[69]. The solution fields corresponding to those in Figure[69] are shown in Figure[70], which clearly demonstrates that the spurious acoustic waves are completely eliminated. The new characteristic boundary conditions are now is being used in all open domain spray simulations. It must be noted that, with the conventional NSCBC [[130]] approach, many more grid points are required to achieve a comparable level of solution fidelity obtained with the new boundary conditions. Based on our experience as reported in previous quarters, at least twice as many grid points in each direction is necessary for the test simulations presented here. Therefore, the new characteristic boundary conditions have led to a significant saving in memory and computational time, thereby substantially contributing toward enhancing the software effectiveness.

During FY06 under the DOE JOULE study, we are targeting to conduct unique multiphase reacting direct numerical simulations of two-stage autoignition and combustion with fully coupled vaporization, turbulence, mixing and chemistry. The simulation will involve 20 million grid points, 30+ species mechanism for n-heptane, and 500,000 spray droplets. A three-dimensional pilot simulation was carried out with 33.5 million grid points and 500,000 n-heptane spray droplets. The mechanism used for n-heptane has 33 species and 58 chemical reactions. The simulations were performed with 256 processors on MPP2 at PNNL. As part of this pilot study, the species were transported but the chemical reaction terms were switched off. From this pilot simulation and previous effective studies, we are confident that the S3D code is fully capable of achieving the JOULE goals.

7.10 Summary of enhancements made in FY2005

7.10.1 Performance enhancements

1. Radiation models: Table[14] shows the overall performance of various radi-

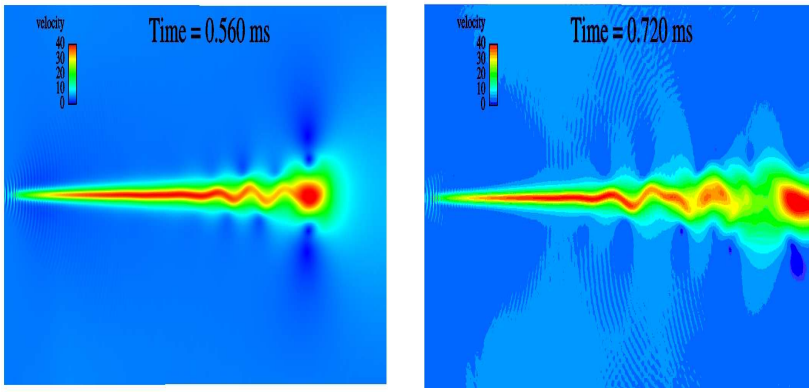

Figure 69: Velocity contours at times 0.56 and 0.72 ms after the start of injection for a 2D spray jet flow with 512×256 grid points. The domain size is 2 cm by 1 cm, and the mass flow rate of fuel spray is 0.125 g/s. The spray droplet diameter is $10\mu m$ and the injection velocity is 100 m/s in x direction. Inflow: hard inflow boundary conditions; outflow: nonreflecting condition based on the NSCBC approach [[130]].

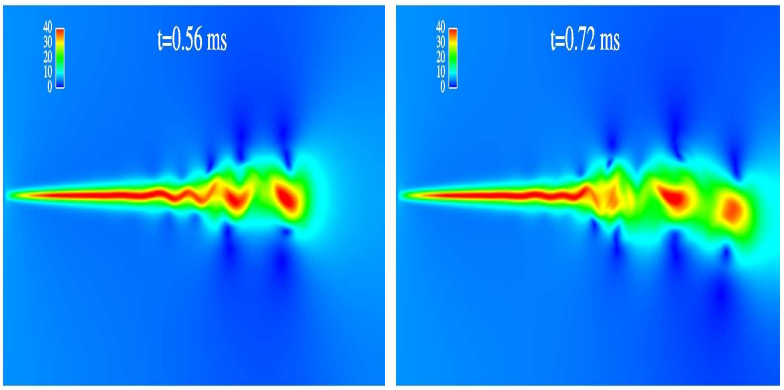

Figure 70: Velocity contours at times 0.56 and 0.72 ms for the spray jet flow. Initial conditions are the same as in Figure[69].The improved inflow and outflow boundary conditions [[129]] were used.

ation models that have been developed during last year. DOM and DTM are the two alternative advanced models that were newly implemented. Between the two models, it was found that DOM-S8, which is now found to be the preferred method, runs faster than DTM-256 by an order of magnitude with comparable accuracy. Note also that DOM adds a reasonable level of computational overhead compared to the simplest OTM, yet with much higher level of solution fidelity, as described in the weak pass cases.

cpus model	OTM	DOM	DTM
1	1.9917e-04	2.4937e-04	1.58190-03
4	1.6688e-04	2.2590e-04	2.3495e-03
16	1.8296e-04	2.5718e-05	3.5221e-03
64	2.2818e-04	4.4564e-04	6.4538e-03
128	3.3907e-04	5.9878e-04	7.5832e-03

Table 14: Performance of various radiation models in terms of CPU time/time-step/grid/processor. Units are seconds.

2. Spray Simulations with Improved Characteristic Boundary Conditions: The newly improved characteristic boundary conditions [[129]] have allowed extremely robust results for high-speed liquid jet simulations without spurious acoustic wave reflections. The level of improvements in the solution fidelity is unprecedented. Based on our previous experience, to achieve a comparable (yet somewhat inferior) level of accuracy using the existing boundary condition treatments, the number of grid points must be doubled in each spatial direction. This implies that the overall savings in the CPU time for the same scale of simulations is up to 75% in 2D problems and 88% in 3D problems. A typical computing cost of S3D with the spray model on Cheetah is $9.318 \times 10^{-9} hours/iteration/gridpoint/processor$.

7.10.2 Scientific enhancements

1. Enhanced Science Capability for Soot and Radiation Models: With the implementation of the advanced radiation and soot models, S3D can now be applied to undertake fundamental studies of transient dynamics of flame extinction and soot formation processes, which will provide valuable guidance in our understanding of the chemistry-turbulence interaction as well as in developing science-based modeling strategies for engineering CFD simulations.

   The OTM approach adopted in the past was found to significantly overestimate the amount of radiative heat loss in many situations of practical relevance, since the model grossly simplifies the absorption of energy by the participating product gases, such as water and carbon dioxide. With

the development of DTM and DOM, we have established two alternative advanced radiation models that will provide substantially improved solution fidelity. In particular, DOM approach was found to the preferred choice in favor of its reasonable computational overhead.

2. Improved Characteristic Boundary Conditions: The development of the improved characteristic boundary conditions to properly account for the acoustic wave reflections at the inflow and outflow boundaries have made a tremendous impact on the applicability of S3D in many complex reacting flow configurations, including turbulent counterflow flames and spray injection/evaporation. S3D is also capable of simulating solid wall boundary conditions with various thermal conditions at the wall.

3. Improved Spray Module: As of October 2004, S3D was not so effective to handle a large number of fuel droplets (with more than 10^5 droplets) on a large number of processors (over 64 processors). The improved spray module is now capable of performing 3D simulations with millions of fuel droplets. The improvement was achieved through the refinement of several key subroutines, such as the data communication and particle tracking between processors, and the use of dynamic allocation of arrays. The new capability will allow us to undertake high-fidelity simulations of spray jet evaporation, mixing, and combustion in a realistic physical scale.

   For a pilot simulation with $512 \times 256 \times 256$ grid points, 500,000 fuel droplets, and a 33-species mechanism for n-heptane, the total memory used was 259 GB. The hard disk requirement is less than 1TB. The computing cost on MPP2 at PNNL is estimated to be $1.4 \times 10^{-7} hours/iteration/grid-point/processor$. For a typical simulation without ignition event, integration up to 200,000 time steps requires 0.853 million CPU hours on MPP2. With ignition and combustion, the actual CPU hours are expected to be further increased.

8 FY04 Reprint: Community Climate System Model

8.1 Overview of the SciDAC CCSM Software Project

The Japanese Earth Simulator (GS40) has challenged the U. S. climate-modeling community and computer industry to accelerate the development of both the models and the computers required to run them, to keep our country competitive in both climate science and policy decisions. The SciDAC project Collaborative Design and Development of the Community Climate System Model for Terascale Computers has the goal of providing a performance portable, state of the science model suitable for use by the climate research community and for climate change studies in support of DOE and other agency missions. The FY2004 development has been aggressive in that it was decided to choose to support the

U. S. participation in the Intergovernmental Panel on Climate Change (IPCC) with simulations of future scenarios under a variety of green house emission scenarios. This required rapid development of additional modeling capability to answer the specific science questions posed by Working Group One as well as aggressive software engineering targeting the available computer hardware so that simulation throughput and schedules could be met.

The SciDAC project Collaborative Design and Development of the Community Climate System Model for Terascale Computers had its initial focus on performance portability. As many of the software modifications and refactoring for performance portability have been completed, more time has been devoted to specific optimizations supporting the IPCC runs and toward development of new science capabilities in the modeling system. The collaborators at the National Center for Atmospheric Research, who coordinate the code development and lead the community of university researchers, have introduced a software engineering discipline to help manage the set of distributed developers and to engineer the code for maximum scientific use with excellent performance on platforms of interest. Supporting the community development process has been a high priority and allowing an open design process with contributions of entire component models by DOE laboratories (LANL in particular) has given new meaning to the interagency support to community model development.

The project has had an active role with several participants in the CCSM Software Engineering Working group. The management of CVS repositories with version control, change review boards, design documents, testing and validation procedures have required significant time and effort from all involved, but there is likely no other way to do it and the group has been very productive under tight deadlines.

The CCSM is a coupled climate system model consisting of atmosphere, ocean, land and sea ice components. Each model has been developed separately, often at different institutions, and runs in coupled mode using a coupler to exchange and regrid outputs and inputs needed for exchange between the components. The entire system is integrated in time typically for 100-200 years taking 10 minute timesteps. It is the long integration, as well as the complexity of the computation, that qualifies climate simulations as a terascale challenge.

The SciDAC Earth System Grid (ESG) project is a national infrastructure supporting the distribution and analysis of simulation model output of the CCSM. It provides for secure automatic migration of data sets between LBNL, ORNL, NCAR and LANL. The project has worked closely with the ESG as major new simulations for the Intergovernmental Panel on Climate Change Fourth Assessment simulations have started production.

The CCSM3.0 code was released in June of 2004 and is being heavily used to study the commitment scenarios of the Intergovernmental Panel on Climate Change (IPCC). Both ORNL and NERSC have joined NCAR in providing machine resources to perform the simulations. Simulations are progressing and runs should conclude in November. A special issue of the *Journal of Climate* is being devoted to description of the CCSM3 and of the results being obtained. A special issue of the *International Journal of High Performacne Computing*

Applications is also planned to describe the parallel algorithms and software design of the CCSM3. The documentation for these publications began in Q4 of FY04.

8.2 CCSM Software Development Path

The Community Climate System Model can be described as a physically based model of the earth's circulations and energy transport. It is based on partial differential equations and physical processes parameterized for the scale of interaction represented in a discrete representation of the atmosphere, ocean, land and sea ice. The software and simulation science is based on the following:

1. Model Equations - Nonlinearly coupled general circulation models representing atmospheric flow over realistic topography and free surface oceanic flow over realistic bathymetry. Moist air thermodynamics includes subgrid scale convective adjustments, cloud physics, full spectral columnar radiation absorption and balance. Thermohaline equations of state with ocean eddy parameterizations of diffusion. Sea ice flows with visco-plastic ice rheology. Fixed ecological and land use areas with consistent energy, biochemical fluxes and plant responses. Fresh water balance through river routing from soil hydrology.

2. Model Boundary Conditions - variable solar input, atmospheric CO2 level, volcanic aerosol concentrations, stratospheric ozone.

3. Initial Conditions - temperature, pressure, velocity, moisture and salinity from historic (or present) climatology.

4. Output - over one hundred 2-D and 3-D fields representing instantaneous states or monthly averages. The volume of output depends on resolution. Typical production runs are 1870-present (validation) and present - 2200 (scenario).

5. Resolution - Atmosphere (spectral T85 =1.4degrees, 10min timestep), Ocean and ice (1 degree).

Three dynamical cores for numerical innovation in the atmosphere are supported. Each provides a unique capability, but each also has deficiencies. The finite volume dycore is being developed to support simulations that must incorporate chemical cycles. (Please see the closing section of this document for updates about the fvd.) The desire is to add these capabilities in order to properly simulate the carbon cycle with a full complement of green house gasses as well as sulfate aerosols and air quality chemicals such as ozone.

The Community Land Model provides the terrestrial component of the coupled system and is being extended to model carbon fluxes with dynamic vegetation processes and realistic land use patterns.

The POP ocean model from LANL is based on free surface formulation using innovative grid structures to get good arctic ocean circulations and sea

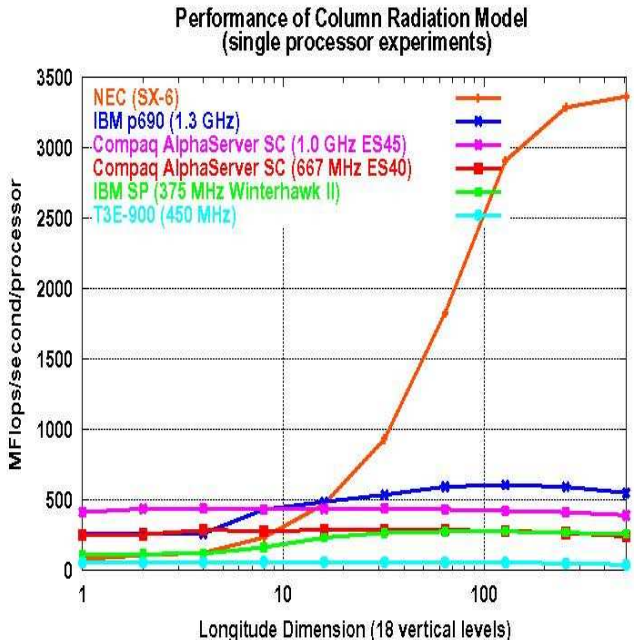

Figure 71: Single processor comparison over platforms for the column radiation model.

ice interactions. A new formulation based on a hybrid vertical coordinate system is being developed in the HYPOP code.

A dynamic sea ice model (CICE) based on visco-plastic rheology is also included in the coupled system.

Vectorization and cache friendly data structures that are adjustable by the user provide much of the performance portability of the CCSM. The project has been able to support the user community and ongoing simulations with a single source code that runs well on all the supported computer platforms. The approach has been put to the test in FY2004 to address the reappearance of vector architectures in the Cray X1 and the NEC SX systems. Simulations are planned on each of these systems. Though it took some work, the layered architecture with utility layer for machine dependent parts minimized the effort in porting to new platforms while maintaining performance on the CCSM standard production platforms at NERSC, NCAR and ORNL. Using a library layer with communication supporting MPI, MPI2, SHMEM, Co-Array Fortran interfaces allows some customization for the specific architecture without affecting the model layer and code readability.

8.3 Performance Metrics

There is no single kernel that can be optimized for the atmospheric code. The dynamics calculation is a spectral transform that involves Fast Fourier Transforms and Legendre function transforms to discretize and solve the semi-implicit equations. Thus, optimized library routines are used where available. The atmospheric physics code represents the column radiation balance and moist processes in the vertical. Long wave and short wave absorption are calculated across a number of spectral bands for the chemical composition of the simulated atmosphere. These calculations are combined with representation of sub-grid scale physics that must be parameterized based on observed, measured data. A column physics benchmark has been developed (PCRM) to indicate the performance of the column radiation within the atmospheric model. The production throughput of component and coupled models is measured in simulated years per day as the primary performance metric.

8.4 Hardware Bottlenecks

Each component model has application characteristics that can utilize specific machine architectural features. The relevant algorithmic characteristics are listed with some comments on sensitivity to machine architectural features.

1. POP ocean code - finite difference with halo-updates and conjugate gradient barotropic solver
2. CSIM ice model - finite difference with halo-updates and incremental remapping advection
3. CAM atmosphere - vector radiation/physics calculations, spectral dynamics, semi-Lagrangian advection with halo updates
4. CLM2 - pointer data structure, required a complete re-write to get an explicit loop in each processing routine to get vector performance
5. CPL6 - coupler utilizes a distributed sparse multiply for interpolation and field regridding in the nxm data transfer
6. Fast network bandwidth allows good distributed performance on spectral methods and data transpositions especially important in CAM
7. Low network latency allows fast collectives and halo updates especially important in POP and CICE.

8.5 Case Study: The Community Atmospheric Model

8.5.1 Scientific Goals and Preparation for the IPCC Simulation Project

The IPCC Simulation project attempts to provide policy makers with a variety of possible climate futures based on carbon dioxide emission scenarios that

correspond to possible economic and technological futures. The standard case is termed 'business as usual' and consists of a 1in atmospheric CO2 with the consequent warming of the earths land and ocean surfaces. Other scenarios are based on carbon stabalization at specified levels and explore the resulting climates.

In order to prepare the CCSM for use in the IPCC Simulation Project a variety of things needed to be added to the code. These included historical volcanic dust and sulfate emissions, GHG emissions, sulfate chemistry, and cloud water prognostics.

8.5.2 Resolution for Regional Impacts

After roughly 10 years at a resolution of T42 (2.8 degrees) (due to lack of improvement in computer interconnection bandwidth), the IBM p690 the resolution to T85 has doubled. This new model allows better resolved regional climates and some effect of hurricanes and extreme weather. The arctic simulation is one region that improved dramatically with the increased resolution. This model also has a better El Nino Southern Oscillation than previous US models.

Long wave radiation balance (and bias) is more physically realistic than CCSM2.

8.5.3 Hardware Example

IBM p690 located at Oak Ridge National Lab consists of 27 nodes each with a 32 way shared memory processor. We use a hybrid MPI OpenMP programming paradigm that allows us to specifically target the machine strengths for distributed memory and shared memory optimization.

8.5.4 Performance Gain

The main code optimizations that were introduced were the load balancing of the chunks, new communicators in atmosphere and with the land model, and a new spectral domain decomposition that allows more fine grain parallelism as well as a smaller memory footprint.

8.5.5 Discussion

With these optimizations the IPCC coupled simulations were run at 5 years /day on the IBM p690. This is allowing completion of the runs within the allotted time.

The same data structures have also been vectorized for the Cray X1 with an observed throughput of 20 years per simulated day at T85 resolution.

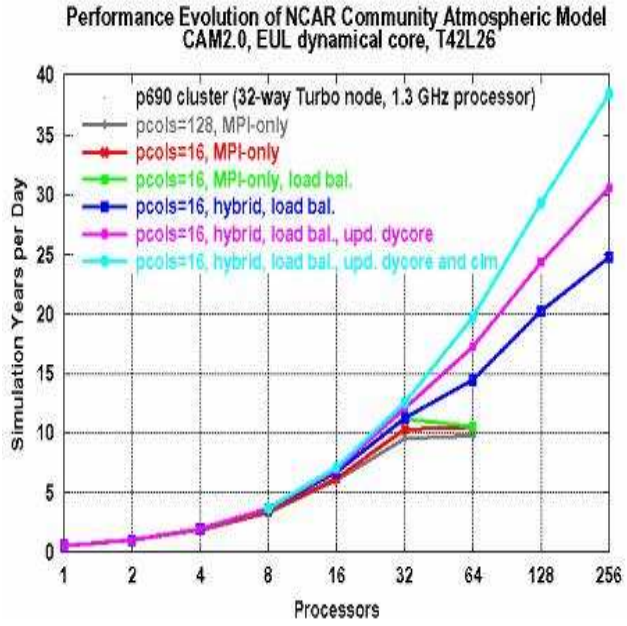

Figure 72: CAM 2.0 in the current fiscal year on the IBM p690 system at ORNL. Distributed and shared memory optimizations are studied here.

8.6 Case Study: The Parallel Ocean Program

8.6.1 Science Goals and Eddy Resolving Simulations

In order to get the surface wind-driven circulation of the oceans correct, it is necessary to resolve mesoscale ocean eddies with spatial scales of 10-50km. Progress in the efficiency of the POP ocean code and in computer capability permitted high resolution simulations of the ocean, resulting in greatly improved representations of ocean circulation. Simulations of the North Atlantic basin at 1/10 degree (4-10km) resulted in very realistic simulations of Gulf Stream separation and subsequent structure around the Grand Banks. The first global simulations at 1/10 degree oceans showed similarly improved representations of the Kuroshio current off Japan, though the Gulf Stream in this simulation was not as realistic as those in the basin-scale simulations, illustrating that eddy-resolving resolution is a necessary, but not sufficient condition for improving ocean simulations.

At climate timescales necessary for IPCC assessments, high resolution simulations like those above remain too computationally expensive. Instead, resolutions of one degree (100km) are used and the effects of mesoscale eddies must be parameterized using computationally intensive schemes like the Gent-McWilliams eddy parameterization.

8.6.2 Software improvements

As mentioned above, flexible blocking data structures have been introduced in version 2.0 of POP. Such blocking enables performance portability by sizing the blocks for cache or vector performance and distributing the blocks more flexibly across a machine for better load balancing and hybrid parallelism. For the ocean, such data structures have the additional advantage of eliminating blocks that are only land.

8.6.3 Performance Gain

For the 1/10 degree model, the new blocking structure on SGI Origin class machines resulted in up to a 30% improvement. Performance at lower resolutions was not substantially improved by the new structure as block sizes were already small and fewer opportunities for land point elimination are available at such coarse resolution.

Performance portability was effectively demonstrated by the performance of POP on the Cray X1 vector machine. Performance of POP on this machine exceeds that of cache-based machines by factors of 6-10. Performance on the X1 at the coarse one-degree resolution even exceeded the Earth Simulator, though the Earth Simulator will probably still be superior at higher resolutions where it can take advantage of the longer vector lengths.

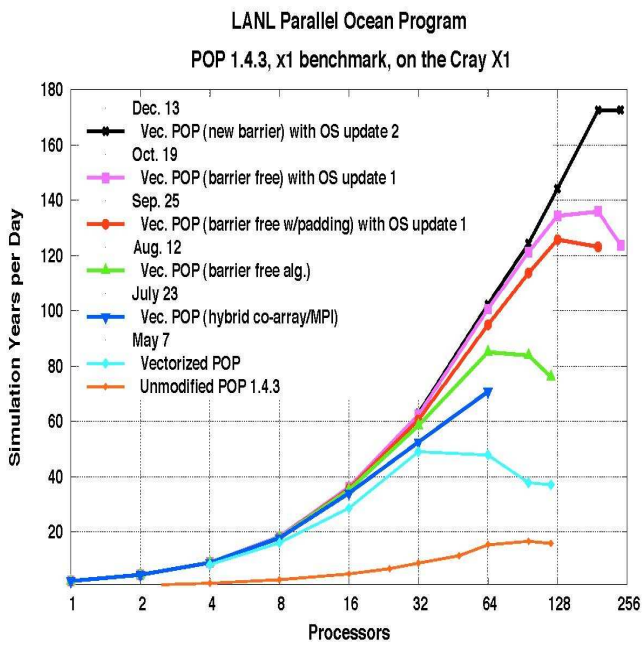

Figure 73: The evolution of performance of the POP code due to software and hardware improvements on the Cray X1 over the last year is shown.

8.6.4 Discussion

The performance of POP on the Cray X1 at one degree resolution typical of IPCC simulations is dramatically better than any other machine available. As the vector length grows for higher resolution simulations, the NEC vector architecture shows significant improvement in efficiency over the 1 degree code. High resolution 1/10 degree simulations will continue to be a focus in the next year with the hope of eventually using such high resolution at climate timescales.

At the end of Q4, the the process of implementing the POP decomposition scheme into the sea ice model started. The outcome of this model change will yield a substantial improvement not only in the ice model, but also a decrease in the amount of information transferred within the coupled system.

8.7 Q3, FY04

At the T42 production resolution, the Community Atmospheric Model reports an improvement from 10 simulated years per day of computation to just over 38 simulated years per day of computation on the IBM p690 system, Cheetah, at ORNL. It is noted that for the T85 higher resolution runs, Cheetah was able to sustain 5 simulated years per day of computation. The Cray X1 system at ORNL can sustain 20 simulated years per day of computation at the same resolution.

Community Atmospheric Model with T42 resolution (Simulated years/day)	old	new	gain
	10	38	280%

For the Parallel Ocean Program, the 1 degree production scale runs on the Cray X1 architecture have evolved from 125 years per day of computation to 193 years per day of computation. The code also reports a 47.82% improvement in the total wall time of the combined barotropic and baroclinic code segments on Cheetah.

Parallel Ocean Program at 1 degree resolution (Simulated years/day)	old	new	gain
	125	193	54.4%

The CCSM3.0 code was released at the end of Q3 for production. This is the coupled, full-component code suite.

8.8 Q4, FY04

It was determined in late Q3 that the Cray X1 should be targeted for the coupled production CCSM code. This was based upon the fact that two key components of the application, CAM and POP, were ported with substantial performance gains in Q3 (as reported above).

There are not new numbers to quantify the gains in performance for Q4. The effectiveness of the software (effort) must be gauged otherwise.

In what follows, it is noted that *verification* implies that all parts (each component model, coupler, I/O, archiving scripts, the production environment) of the production code are functioning and that subsequent runs of the same problem instance produce the same output.

Validation is harder to achieve and hence more time consuming. It implies that the code is producing the correct model climate. Code that has been validated has necessarily been verified.

The substantial achievements in Q4:

- The production code has run IPCC scenarios essentially non-stop on the IBM systems at ORNL and NERSC.
- The production code port to the Cray X1 has completed verification.
- The production code port to the Cray X1 began but has not completed validation on the Cray X1.
- The CAM3 atmospheric model in conjunction with the CLM3 land model has cleared validation .

  ```
   A high resolution AMIP experiment is being developed
  for this case.
  ```

- The experimental branches of the code that are adding dynamic vegetation, land carbon accounting and a nitrogen cycle have been vectorized and ported to the Cray X1.
- The finite volume dynamical core of the atmosphere has been vectorized and ported to the Cray X1.

  ```
  This enables effective use of the atmospheric model with
  coupled tropospheric and stratospheric chemistry being
  exercised in a prototyping branch. As such, the change
  is not designed for standard IPCC runs.
  ```

- The new POP2.0 ocean code was integrated into the CCSM framework.

  ```
  This is the first (public) of several accepted
  deliverables for DOE's Climate Change Prediction
  Program.
  ```

References

[1] T. H. Stix, Waves in Plasmas, American Institute of Physics, New York, NY. (1992).

[2] E. F. Jaeger, L. A. Berry, J. R. Myra, et al., Phys. Rev. Lett. 90, 195001-1 (2003).

[3] J. R. Myra, L. A. Berry, D. A. DIppolito, and E. F. Jaeger, Phys. Plasmas 11, 1786 (2004).

[4] H. Biglari, P. H. Diamond, and P. W. Terry, Phys. Fluids B 2, 1 (1990).

[5] R. Cesario, C. Castaldo, V. Pericoli-Ridolfini, et al., "Recent Results of the Ion Bernstein Wave Heating Experiment on FTU," in Proceedings of the 13th Topical Conference on Radio Frequency Power in Plasmas, 1999 Annapolis, MD, Edited by S. Bernabei and F. Paoletti (American Institute of Physics, New York, 1999), p. 100.

[6] ITER Physics Basis: Plasma Confinement and Transport, Nucl. Fusion 39, 2175 (1999).

[7] M. D. Carter, D. A. Rasmussen, P. M. Ryan, et al., Nucl. Fusion 36, 209 (1996).

[8] M. Brambilla and T. Krucken, Nucl. Fusion 28, 1813 (1988).

[9] E. F. Jaeger, D. B. Batchelor and D. C. Stallings, Nucl. Fusion 33, 179 (1993).

[10] E. F. Jaeger, L. A. Berry, E. DAzevedo, D. B. Batchelor, and M. D. Carter, Phys. Plasmas 8, 1573 (2001).

[11] D. N. Smithe, Plasma Phys. Controlled Fusion 31, 1105 (1989).

[12] R. W. Harvey and M. G. McCoy, in Proceedings of the IAEA Technical Committee Meeting on Advances in Simulation and Modeling of Thermonuclear Plasmas (IAEA, Montreal, 1992), available through USDOC, NTIS No. DE9300962.

[13] V. S. Chan, S. C. Chiu, and Y. A. Omelchenko, Phys. Plasmas 9, 501 (2002).

[14] M. Brambilla, Kinetic Theory of Plasma Waves, Clarendon Press, Oxford, (1998).

[15] J. C. Wright, P. T. Bonoli, M. Brambilla, et al., phys. Plasmas 11, 2473 (2004).

[16] E. F. Jaeger, L. A. Berry, E. DAzevedo, et al., Phys. Plasmas 9, 1873 (2002).

[17] C. F. Kennel and F. Engelmann, Phys. Fluids 9, 2377 (1966).

[18] R.J. Dumont, C.K. Phillips and D.N. Smithe, "ICRF wave propagation and absorption in plasmas with non-thermal populations," in Controlled Fusion and Plasma Physics, 26B, paper P-5.051, 29th EPS Conference, Montreux, Jun. 17-21, (2002).

[19] R.J. Dumont, C.K. Phillips, and D.N. Smithe, "Effects of non-Maxwellian Plasma Species on ICRF Propagation and Absorption in Toroidal Magnetic Confinement Devices," in Radio Frequency Power in Plasmas: 15th Topical Conference, AIP Conf. Proc. 694, p. 439, American Institute of Physics, New York, NY. (2003).

[20] R. J. Dumont, C. K. Phillips, and D. N. Smithe, "Effects of non-Maxwellian Species on Electromagnetic Wave Propagation and Absorption in Magnetically Confined Plasmas," Submitted to Phys. Rev. Lett. (2004).

[21] C. K. Phillips, M. G. Bell, R. E. Bell, et al., Nucl. Fusion 40, 461 (2000).

[22] D. J. Campbell, et al., Phys. Rev. Lett. 60, 2148 (1983).

[23] B. D. Fried and S. D. Conte, The Plasma Dispersion Relation, Academic Press, New York, NY. (1961).

[24] T. K. Mau, C. C. Petty, M. Porkolab, and W. W. Heidbrink, in Radio Frequency Power in Plasmas: 13th Topical Conference, AIP Conf. Proc. 485, p. 148, American Institute of Physics, New York, NY. (1999).

[25] A. L. Rosenberg, J. E. Menard, J. R. Wilson, et al., Phys. Plasmas 11, 2441 (2004).

[26] Laub, A., *Matrix Analysis for Scientists and Engineers*, SIAM Publishers, Philidelphia, 2005.

[27] Yosida, K., *Functional Analysis*, Spriner-Verlag, New York, 1978.

[28] Work on the latest version of LAMMPS was funded in part by the US Department of Energy's Genomics:GTL program (www.doegenomestolife.org) under the project "Carbon Sequestration in Synechococcus Sp.: From Molecular Machines to Hierarchical Modeling" (www).

[29] LAMMPS WWW page: `www.cs.sandia.gov/ sjplimp/lammps.html`.

[30] S. J. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics," J Comp Phys, 117, 1-19 (1995).

[31] S. J. Plimpton, R. Pollock, M. Stevens, "Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations," in Proc of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN (March 1997).

[32] Work on the latest version of LAMMPS was funded in part by the US Department of Energy's Genomics:GTL program (www.doegenomestolife.org) under the project "Carbon Sequestration in Synechococcus Sp.: From Molecular Machines to Hierarchical Modeling" (www).

[33] LAMMPS WWW page: www.cs.sandia.gov/ sjplimp/lammps.html.

[34] Pizza.py WWW page: www.cs.sandia.gov/ sjplimp/pizza.html.

[35] P. S. Crozier, "Need to get this title and journal citation", submitted to J Name. (June 2005).

[36] X. Cheng, J. B. Lechman, A. F. Barbero, G. S. Grest, H. M. Jaeger, G. S. Karczmar, M. E. Mobius, and S. R. Nagel, "Onset of Three-Dimensional Shear in Granular Flow", submitted to Phys. Rev. Lett. (July 2005).

[37] International Linear Collider Communication, `http://www.interactions.org/linearcollider/` (2005).

[38] TESLA Technical Design Report, `http://tesla.desy.de/new_pages/TDR_CD/start.html`.

[39] Y. Sun and N. Folwell and Z. Li and G. Golub, High precision accelerator cavity design using the parallel eigensolver Omega3P, Proceedings of the 18th Annual Review of Progress in Applied Computational Electromagnetics, (2002).

[40] Y. Sun, The filter algorithm for solving large-scale eigenproblems from accelerator simulations, PhD Thesis, Stanford University (2003).

[41] Din-Kow Sun and Jin-Fa Lee and Zoltan Cendes, Construction of Nearly Orthogonal Nedelec Bases for Rapid Convergence with Multilevel Preconditioned Solvers, SIAM Journal on Scientific Computing, V23, 4, 1053-1076 (2001).

[42] J.-C. Nédélec, Mixed finite elements in $\mathbf{R}^3$, Numerische Mathematik, v35, 3, 315-341 (1980).

[43] J. Jin, The Finite Element Method in Electromagnetics, 2nd Edition, John Wiley & Sons Inc., New York, NY (2002).

[44] L. Lee and L. Ge and M. Kowalski and Z. Li and C. Ng and G. Schussman and M. Wolf and K. Ko, Solving Large Sparse Linear Systems in End-to-end Accelerator Structure Simulations, Proceedings of the 18th International Parallel and Distributed Processing Symposium (2004).

[45] L. Lee and Z. Bai and W. Gao and L. Ge and P. Husbands and M. Kowalski and X. Li and Z. Li and C. Ng and C. Yang and E. G. Ng and Kwok Ko, Modeling RF Cavities with External Coupling, SIAM Conference on Computational Science and Engineering, Orlando, Florida (2005).

[46] K. Ko and N. Folwell and L. Ge and A. Guetz and A. Kabel and V. Ivanov and L.-Q. Lee and Z. Li and I. Malik and W. Mi and C.-K. Ng and Y. Sun and R. Uplenchwar and M. Wolf, Electromagnetic Systems Simulation - Prototyping through Computation, SciDAC Report (2004).

[47] K. Ko and N. Folwell and L. Ge and A. Guetz and V. Ivanov and L.-Q. Lee and Z. Li and I. Malik and W. Mi and C.-K. Ng and M. Wolf, Electromagnetic Systems Simulation - From Simulation to Fabrication, SciDAC Report (2003).

[48] Lixin Ge and Lie-Quan Lee and Zenghai Li and Cho Ng and Kwok Ko and Yunhua Luo and Mark Shephard, Adaptive Mesh Refinement for High Accuracy Wall Loss Determination in Accelerating Cavity Design, Proceedings of the Eleventh Biennial IEEE Conference on Electromagnetic Field Computation (2004).

[49] Chao Yang and Weiguo Gao and Z. Bai and Xaoye Li and Lie-Quan Lee and Parry Husbands and Esmond Ng, An Algebraic Sub-structuring Method for Large-scale Eigenvalue Calculation, LBNL-55050, Lawrence Berkeley National Laboratory, Berkeley, California (2004).

[50] Z. Li and N. Folwell and L. Ge and A. Guetz and V. Ivanov and M. Kowalski and L.-Q. Lee and C.-K. Ng and G. Schussman and R. Uplenchwar and M. Wolf and L. Xiao and K. Ko, High Performance Computing in Accelerator Structure Design and Analysis, Proceedings of the 8th International Computational Accelerator Physics Conference (2004).

[51] Zenghai Li and Nathan T. Folwell and Lixin Ge and Adam Guetz and Valentin Ivanov and Marc Kowalski and Lie-Quan Lee and Cho-Kuen Ng and Greg Schussman and Ravindra Uplenchwar and Michael Wolf and Kwok Ko, X-band Linear Collider R&D in Accelerating Structures Through Advanced Computing, Proceedings of the 9th European Particle Accelerator Conference (2004).

[52] R. B. Lehoucq and D. C. Sorensen and C. Yang, ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM, Philadelphia, PA (1998).

[53] Z. Bai and J. Demmel and J. Dongarra and A. Ruhe and H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, PA (1998).

[54] Zhaojun Bao and Yangfeng Su, "SOAR: A Second-order Arnoldi Method for the Solution of the Quadratic Eigenvalue Problem," SIAM Journal on Matrix Analysis and Applications, v26, 3, 640-659 (2005).

[55] Anshul Gupta, WSMP: Watson Sparse Matrix Package, `http://www-users.cs.umn.edu/~agupta/wsmp.html`.

[56] MUMPS: a Multifrontal Massively Parallel sparse direct Solver, `http://graal.ens-lyon.fr/MUMPS/`.

[57] Xiaoye S. Li and James W. Demmel, "$SuperLU_{DIST}$: A scalable distributed-memory sparse direct solver for unsymmetric linear systems," ACM Trans. Math. Softw., v29 (0098-3500), 2, 110-140 (2003).

[58] George Karypis and Vipin Kumar, METIS, http://www-users.cs.umn.edu/k̃arypis/metis/metis/index.html.

[59] Hong-Jun Kim and Timothy J. Tautges and John Uicker, Unstructured Parallel Mesh Generation By Distributing and Partitioning Solid Geometry Model, SIAM Conference on Computational Science and Engineering, Orlando, Florida (2005).

[60] Z. Li and N. Folwell and K. Ko and R.J. Loewen and E.W. Lundahl and B. McCandless and R.H. Miller and R.D. Ruth and M.D. Starkey and Y Sun and J.W. Wang, Design of the JLC/NLC RDDS Structure Using Parallel Eigensolver Omega3P, Intertional Linac Conference, Monterey, California (2000).

[61] David Keyes, Shape Optimization for Accelerator Structures, Accelerator Science SciDAC Meeting, Fermi Lab (2004).

[62] Michael Wolf and Ali Pinar and Adam Guetz and Kwok Ko, Using Parallel Mesh Partitioning Strategies to Improve the Performance of Tau3P, An Electromagnetic Field Solver, Parallel Processing for Scientific Computing (2004).

[63] R. B. Bird and W. E. Stewart and E. N. Lightfoot, Transport Phenomena, Wiley, New York, New York (1960).

[64] W. K. Bushe and H. Steiner, Phys. Fluids, v11, 7, 1896-1906 (1999).

[65] C. M. Cha and H. Pitsch, Combust. Theory Modelling, v6, 425-437 (2002).

[66] J. H. Chen and H. G. Im, Proc. Combust. Inst., v28, 211-218 (2000).

[67] J. H. Chen and E. R. Hawkes and R. Sankaran and S. D. Mason and J. Sutherland and C.A. Kennedy, Sandia Internal Report (in preparation), Sandia National Laboratory (2005).

[68] G. Dixon-Lewis, "Flame Structure and Flame Reaction Kinetics," Proc. Roy. Soc, A., v307, 111-135 (1968).

[69] T. Echekki and J. H. Chen, Combust. Flame, v134, 169-191 (2002).

[70] A. Ern and V. Giovangigli, "Thermal Diffusion Effects in Hydrogen-Air and Methane-Air Flames," Combust. Theory Modelling, v2, 349-372 (1998).

[71] D. Geyer and A. Kempf and A. Dreizler and J. Janicka, Proc. Combust. Inst., v30 (2004) (to appear).

[72] V. Giovangigli, Multicomponent Flow Modeling, Birkhauser, Boston, MA. (1999).

[73] E. R. Hawkes and J.H. Chen, Combust. Flame, v3, 138, 242-258 (2004).

[74] R. Hilbert and F. Tap and H. El-Rabii and D. Thévenin, Prog. Energy Combust. Sci, v30, 61-117 (2004).

[75] , J. O. Hirschfelder and C. F. Curtis and R. B. Bird, The Molecular Theory of Liquids and Gases, Wiley, New York, NY. (1954).

[76] J. C. Hewson and A. Kerstein, Combust. Sci and Tech., v174, 35-66 (2002).

[77] A. Karpetis and R. Barlow, Proc. Combust. Inst., v30 (2004) (to appear).

[78] C. A. Kennedy and M. H. Carpenter and R.M. Lewis, Appl. Num. Math, v14, 4, 397-433 (1994).

[79] A. Kerstein, J. Fluid Mech., v392, 277-334 (1999).

[80] A. Y. Klimenko and R.W. Bilger, Prog. Energy Combust. Sci., v25, 595-687 (1999).

[81] A. Kronenburg and R. W. Bilger, Combust. Sci. Tech., v166, 175-194 (2001).

[82] A. Kronenburg and R. W. Bilger, Combust. Sci. Tech., v166, 195-227 (2001).

[83] A. Kronenburg and A. E. Papoutsakis, Proc. Combust. Inst., v30 (2004) (to appear).

[84] J. Li and Z. Zhao and A. Kazakov and F. L. Dryer, Int. J. Chem. Kinetics (2005) (to appear).

[85] S. Mahalingam and J.H. Chen and L. Vervisch, Combust. Flame, v102, 285-297 (1995).

[86] S. Mitarai and G. Kosĺy and J. J. Riley, Physics of Fluids, v15, 12, 3856-3866 (2003).

[87] S. Mitarai and G. Kosĺy and J. J. Riley, Combust. Flame, v137, 306-319 (2004).

[88] Y. Mizobuchi and S. Tachibana and J. Shinjo and S Ogawa and T. Takeno, Proc. Combust. Inst., v29, 2009-2015 (2002).

[89] Y. Mizobuchi and J. Shinjo and S Ogawa and T. Takeno, Proc. Combust. Inst., v30 (2004) (to appear).

[90] A. Kronenburg and S. Navarro-Martinez, Physics of Fluids (2005) (submitted)

[91] C. Pantano, J. Fluid Mech. (2004) (to appear).

[92] N. Peters, Turbulent Combustion, Cambridge University Press, New York, NY. (2000).

[93] H. Pitsch and C. M. Cha and S. Fedotov, Combust. Theory Modelling, v7, 317-332 (2003).

[94] H. Pitsch and H. Steiner, Physics of Fluids, v12, 10, 2541-2554 (2000).

[95] T. Poinsot and S. Candel and A. Trouvé, Prog. Energy Combust. Sci., v21, 531-576 (1996).

[96] T. J. Poinsot and S. K. Lele, "Boundary Conditions for Direct Simulations of Compressible Viscous Flows," J. Comp. Phys., v101, 1, 104-139 (1992).

[97] S. B. Pope, Prog. Energy Combust. Sci., v11, 119-192 (1985).

[98] R. Sankaran and H. G. Im and E. R. Hawkes and J. H. Chen, Proc. Combust. Inst., v30 (2004) (to appear).

[99] P. Sripakagorn and S. Mitarai and G. Kosly and H. Pitsch, J. Fluid Mech. (2004) (submitted).

[100] J. Sutherland and P. J. Smith and J.H. Chen, Combust. Theory and Modeling (2004) (to appear).

[101] J. Sutherland and C. A. Kennedy, J. Comp. Phys., v191, 502-524 (2004).

[102] L. Vervisch and T. Poinsot, Annu. Rev. Fluid Mech., v30, 655-691 (1998).

[103] J. Warnatz, U. Maas, R. W. Dibble, Combustion , 1st ed. , Springer, Berlin, DE. (1996).

[104] R. A. Yetter and F. L. Dryer and H. Rabitz, Comb. Sci. Tech., v79, 97-128.

[105] S. B. Pope, Prog. Energy Combust. Sci., v11, 119-195 (1984).

[106] S. B. Pope, Proceedings of Sixth International Workshop on the Measurement and Computation of Turbulent Nonpremixed Flames, 2002.

[107] A. N. Karpetis and R. S. Barlow, Proc. Combust. Inst., v30, 665-672 (2005).

[108] C. Pantano, S. Sarkar and F. A. Williams, J. Fluid Mech., v481, 291-328 (2003).

[109] Subramanya, R. and Reddy, "Sandia DNS code for 3D compressible flows - Final Report," PSC-Sandia-FR-3.0, Pittsburgh Supercomputing Center, PA. (2000).

[110] Lele, S. K., Compact finite difference schemes with spectral-like resolution," J. Comp. Phys. 103:16-42 (1992).

[111] Kennedy, C. A. and Carpenter, M. H., A comparison of several numerical methods for the simulation of compressible shear layers," Appl. Num. Math. v14, 4, 397-433 (1994).

[112] Kennedy, C. A., Carpenter, M. H. and Lewis, R. H., Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations, Appl. Num. Math. v35, 3, 177-219 (2000).

[113] Kennedy , C. A. and Carpenter, M. H. Additive Runge-Kutta schemes for convection-diffusion reaction equations, v44, 139-181 (2003).

[114] Poinsot, T. and Lele, S. K., Boundary conditions for direct simulations of compressible viscous flows, J. Comp. Phys. v101 , 104-129 (1992).

[115] Baum, M., Poinsot, T. and Thevenin, D., Accurate boundary conditions for multi-component reactive flows, J. Comp. Phys. v116, 2, 247-261 (1995).

[116] Kee, R. J., Rupley, F. M. and Miller, J. A., CHEMKIN-II: a Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics," Sandia National Laboratories report SAND89-8009 (1990).

[117] Parashar, M., et al., A common data management infrastructure for parallel adaptive algorithms for PDE solutions, Proc. Supercomputing 1997.

[118] Parashar, M. and Browne, J. C., A Computational Infrastructure for Parallel Adaptive Algorithms, IEEE Computational Science (1999).

[119] Chandrasekhar, S., Radiative Transfer, Dover, New York, NY (1960).

[120] Lockwood, F. C. and Shah, N. G., Proc. Combust. Inst., v18, 1405-1414 (1981).

[121] Moss, J. B., Stewart, C. D. & Young, K. J., Combust. Flame, v101, 491-500 (1995).

[122] Brookes, S. J. and Moss, J. B., Combust. Flame, v116, 486-503 (1999).

[123] Mashayek, F., J. Fluid Mech. v367, 163-203 (1998).

[124] Bauman, S. and Rutland, C. J., Spray Modeling With Adaptive Mesh Refinement, 8th SIAM Intl. Conf. on Numerical Combustion, Amelia Island, FL, March 2000.

[125] Wang, Y. and Rutland, C. J., On the combustion of normal heptane fuel droplets in isotropic turbulence with DNS, The Third Joint Meeting of the U.S. Sections of the Combustion Institute, Paper E-39, Chicago, IL, March 2003.

[126] Najm, H. N. et al., A Computational Facility for Reacting Flow Science, SciDAC Project, `http://cfrfs.ca.sandia.gov/`.

[127] A. Malony and S. Shende, Performance Technology for Complex Parallel and Distributed Systems, Proc. Third Austrian-Hungarian Workshop on Distributed and Parallel Systems, DAPSYS 2000, "Distributed and Parallel Systems: From Concepts to Applications," (Eds. G. Kotsis and P. Kacsuk) Kluwer, Norwell, MA, pp. 37-46 (2000). Also, `http://www.cs.uoregon.edu/research/paracomp/tau/tautools/`.

[128] G.A. Blaisdel and E.T. Spyropoulos and J.H. Qin, The effect of the formulation of nonlinear terms on aliasing errors in spectral methods, Applied Numerical Mathematics, v21, 207-219, (1996).

[129] C.S. Yoo and Y. Wang and A. Trouve and H.G. Im, "Characteristic boundary conditions for direct numerical simulations of turbulent counterflow flames," Combust. Theory Modelling, in press, (2005).

[130] T.J. Poinsot, S.K. Lele, Boundary Conditions for Direct Simulations of Compressible Viscous Flows, Journal of Computational Physics, 101:104-139, (1992).

[131] J.C. Sutherland, C.A. Kennedy, Improved boundary conditions for viscous, reacting, compressible flows, Journal of Computational Physics, 191:502-524, (2003).