US DOE SC OASCR
FY07 Joule Software Metric

October 19, 2007

Contents

Credits **5**

The Joule System **7**
0.1 OASCR's FY07 Joule Goals 7
0.2 Related Quarterly Tasks 8
0.3 A context for Joule 9

Discussion of Q2, Q4 Results **11**
0.4 Results 11
0.4.1 `jaguar.ccs.ornl.gov` 11
0.4.2 Chimera 13
0.4.3 GTC-S 15
0.4.4 S3D 17

Chimera Discussion **21**
0.5 Objective and Expected Significance 21
0.6 Background and Approach 22
0.6.1 Laying the Foundation 22
0.6.2 A Staged Approach 23
0.7 CHIMERA Overview 24
0.8 CHIMERA's Hydrodynamics Module 27
0.8.1 Solving the Hydrodynamics Equations 27
0.8.2 Solving the Poisson Equation 32
0.9 CHIMERA's Transport Module 34
0.9.1 Overview 34
0.9.2 Neutrino Boltzmann Equation 34
0.9.3 Neutrino Boltzmann Equation in Terms of Independent Variables $(t, m, a\epsilon, \mu)$ 39
0.9.4 Moment Equations 39
0.9.5 Flux-Limiting 41
0.9.6 Matter–Neutrino Energy–Momentum Exchange 45
0.9.7 Einstein Equations 48
0.9.8 Matter–Neutrino Lepton Number Exchange 50
0.9.9 Summary of Equations 51

0.9.10 The equations presently used in CHIMERA 52
0.9.11 Differencing of the MGFLD Equations 53
0.9.12 Solution of the MGFLD Equations 63
0.10 CHIMERA's Network and Equation of State Modules 70
0.10.1 Nuclear Statistical Equilibrium 71
0.10.2 The High Density Equation of State and Nuclear Matter . . . 73
0.10.3 Non-Equilibrium Regions and Thermonuclear Reaction Networks 73
0.11 CHIMERA Problem Statement and Results 87
0.11.1 Q2 87
0.11.2 Q4 88
0.12 CHIMERA Verification 88
0.13 Toward the Petascale and Higher-Fidelity 3D Simulations 89
0.13.1 Architectural Perspective 89
0.13.2 Science Overview 89
0.13.3 Effective Utilization of Multi-Core Sockets 90
0.13.4 New Communication Strategies 91
0.13.5 Parallel I/O & Analysis Tools 92
0.13.6 Data Movement and Workflow Management 93
0.13.7 Spatial Grid 93
0.13.8 Three-Dimensional Gravity 94
0.13.9 Magnetohydrodynamics 95
0.14 Timetable 95
0.15 Results from Ongoing 2D Simulations 96
0.16 Planned 3D Simulations: Core Collapse Supernova Mechanism and Observables 97

GTC-S Discussion **105**
0.17 GTC-S: Gyrokinetic Simulation of Global Turbulent Transport in Fusion Plasmas 105
0.17.1 Introduction 105
0.18 General Geometry Gyrokinetic Particle Simulation Model 107
0.18.1 Coordinate system and mesh construction 107
0.18.2 Gyrokinetic transformation 109
0.18.3 Basic equations 112
0.19 Parallel Model and Optimizations 113
0.19.1 Parallel Model 113
0.19.2 Optimizations 114
0.20 Tasks Undertaken to Achieve Joule Milestones 115
0.20.1 Performance Metric 115
0.20.2 Problem Statement 116
0.20.3 Initial Performance Results at the End of Q2 116
0.20.4 Performance Results at the End of Q4 117
0.21 New High Performance Parallel I/O in GTC-S 117
0.21.1 GTC-S Incorporation 120
0.21.2 Initial Performance 120

0.22 Summary and Future Plans 120
0.23 Acknowledgement 122

S3D Description **123**
0.24 S3D : Sandia INCITE Team 123
0.25 Q2 Problem Information 123
0.25.1 Background and Motivation 123
0.25.2 Approach and Goals 123
0.25.3 Problem Configuration 124
0.25.4 Reduced Chemical Mechanism for Lean Premixed Methane-Air Flames 125
0.25.5 Mechanism Reduction Methodology 126
0.25.6 Mechanism Validation 127
0.25.7 Numerical Method 128
0.25.8 Q2 Benchmark Run Description 129
0.25.9 Effect on flame thickness 130
0.25.10 Tracer particle 133
0.25.11 Flame element tracking 134
0.25.12 Data Analysis 135
0.26 Q4 Problem Information 135
0.26.1 Ethylene chemical mechanism 135
0.26.2 Configuration and Initial Conditions 136
0.26.3 Results and Discussion 137
0.27 S3D Performance 141
0.27.1 Introduction 141
0.27.2 Improving the node performance of S3D. 145

REPRINT of FY05 S3D INCITE OASCR Joule software assertion **149**
0.28 S3D : A Direct Numerical Simulation of Chemical Combustion 149
0.28.1 Introduction 149
0.28.2 Background and Motivation 150
0.28.3 Research Goals 152
0.28.4 Computational Approach 155
0.28.5 Physical Configuration 155
0.28.6 Physical and Numerical Parameter Selection Considerations . 156
0.28.7 Formulation 159
0.28.8 Code Overview 163
0.28.9 S3D Software Performance and Improvements 165
0.28.10 Test problem description 165
0.28.11 Scalar profiling and performance improvements 166
0.28.12 Evaluation of parallel scaling and communication overheads . 170
0.28.13 Evaluation of memory limitations 173
0.28.14 Q3 Progress and Dynamics 173
0.28.15 Scientific Insights from DNS Simulations 175
0.28.16 Knowledge discovery from terascale simulated combustion data 180

0.28.17 Prospects for future combustion simulations on petascale platforms 181
0.28.18 Code Improvements made in FY05-Q4 183

Bibliography **187**

Credits

Application Credits

`Chimera :` Anthony Mezzacappa (mezzacappaa@ornl.gov), Physics Division, Theoretical Astrophysics, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, MS - 6354 Oak Ridge, TN 37831-6354, U.S.A

`GTC-S :` Wei-li Lee (wwlee@pppl.gov), Theory Department, Princeton Plasma Physics Laboratory, P.O. Box 451, Princeton, NJ 08543-0451, U.S.A

`S3D :` Jacqueline Chen (jhchen@sandia.gov), Combustion Research Facility, Sandia National Laboratory, P.O. Box 969, Livermore, California 94551, U.S.A

Technical Team : (chimera) John Blondin (North Carolina State), Stephen Bruenn (Florida Atlantic), Raphael Hix (ORNL), Bronson Messer (NCCS ORNL), (gtc-s) Mark Adams (Columbia), Stephane Ethier (PPPL), Scott Klasky (NCCS ORNL), Weixing Wang (PPPL), (s3d) Evatt Hawkes (New South Wales), Chung Law (Princeton), David Lignell (SNL), Tianfeng Lu (Princeton), Ramanan Sankaran (NCCS ORNL), Chun Yoo (SNL)

Additional Credits

David Bailey (LBNL), Bronis de Supinski (LLNL), Jeffrey Larkin (Cray Inc.), Ricky Kendall (NCCS ORNL), Doug Kothe (NCCS ORNL), John Mellor Crummey (Rice), Shirley Moore (Tennessee, ICL), Kenneth Roche (ORNL), Sameer Shende (Oregon)

DOE Program Contacts

Daniel Hitchcock, daniel.hitchcock@science.doe.gov
Frederick Johnson, fjohnson@er.doe.gov
Michael Strayer, michael.strayer@science.doe.gov

Point of Contact

Kenneth Roche (rochekj@ornl.gov), 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 37831-6173, U.S.A

The Joule System

0.1 OASCR's FY07 Joule Goals

The Office of Advanced Scientific Computing Research (OASCR)[1] has the following two annual measures that it tracks quarterly:

1. **(SC GG 3.1/2.5.1)** Focus usage of the primary supercomputer at the National Energy Research Scientific Computing Center (NERSC) on capability computing. Percentage of the computing time used that is accounted for by computations that require at least 1/8 of the total resource. *FY07: time used is at least 40%*

2. **(SC GG 3.1/2.5.2)** 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)[2] effort. *FY07: efficiency measure is 100%*

Asserting compliance with these metrics is a critical hurdle each fiscal year for the success of DOE's open science computing effort. This document presents the results for the computational effectiveness capability.

[1] http://www.sc.doe.gov/ascr/About/about.html
[2] http://www.scidac.gov

0.2 Related Quarterly Tasks

This is a year long effort requiring quarterly updates. The general outline of tasks for exercising the software metric are presented by fiscal quarter here.

Q1 Tasks (`deadline: Decemeber 31`)
Identify a subset of candidate applications to be investigated on DOE SC supercomputers. Management (DOE SC and laboratory) decides a short list of applications and computing platforms to be exercised. The Advanced Scientific Computing Advisory Committee (ASCAC) approves or rejects the list. The Q1 milestone is satisfied when a short list of applications and machines is approved.

Q2 Tasks (`deadline: March 31`)
Problems to study on the target machines are determined. The science capability and computational performance of the implementation are benchmarked on the target machines for the defined problems, problem instances. The Q2 milestone is satisfied when benchmark data is collected and explained. In the case that an application is aiming to achieve a new result, the Q2 milestone is satisfied by providing a detailed discussion of current capability, a discussion of why the capability is insufficient, and a description of the new capability being developed.

Q3 Tasks (`deadline: June 30`)
The application software is enhanced for efficiency, scalability, science capability, etc. The Q3 milestone is satisfied when the status of each application is reported at the Q3 deadline. Corrections to Q2 problem statements are submitted during this quarter.

Q4 Tasks (`deadline: September 30`)
Enhancements to the application software continue as in Q3. The enhancements are stated and demonstrated on the machines used to generate the baseline information. A comparative analysis of the Q2 and Q4 data is summarized and reported. The Q4 milestone is satisfied by asserting that the enhancements made to the application software are in accordance with the efficiency measure and type of enhancement -efficiency, scalability, or new result.

0.3 A context for Joule

Here is some recent history. The U.S. Office of Management and Budget (OMB)[3] oversees the preparation and administration of the President's budget, evaluates the effectiveness of agency programs, policies and procedures, assesses competing funding demands across agencies, and sets the funding priorities for the federal government. In some sense, the buck ($) stops with OMB!

Public Authorizations
PL 95-91, "Department of Energy Organization Act"
PL 103-62, "Government Performance and Results Act"

OMB has the power of audit and exercises this right annually for each agency. According to the *Government Performance and Results Act of 1993 (GPRA)*, federal agencies are required to develop three planning and performance documents:

1. Strategic Plan: broad, three year outlook
2. Annual Performance Plan that is incorporated into the annual budget request: focused, one year outlook of annual goals and objectives; "what results can the agency produce for the taxpayers money?"
3. Performance and Accountability Report: an annual report about the past fiscal year performance; "what results did the agency produce for the taxpayer's money?"

OMB uses its *Performance Assessment Rating Tool (PART)* to perform evaluations. PART has seven worksheets for seven types of agency functions. The function of Research and Development (R&D) programs is included. R&D programs are assessed upon the following criteria:

- does the R&D program perform a clear role?
- has the program set valid long term and annual goals?
- is the program well managed?
- is the program achieving the results set forth in its GPRA documents?

In FY2003, the Department of Energy Office of Science (DOE SC) worked directly with OMB to come to a concensus on an appropriate set of performance measures consistent with PART requirements. The scientific performance expectations of these requirements reach the scope of work conducted at the national laboratories. The *Joule* system emerged from this interaction. Joule enables the chief financial officer and senior DOE management to track annual performance on a quarterly basis. Joule scores are reported as "goal met" (*green light* in PART), "goal partially met" (*yellow light* in PART), and "goal not met" (*red light* in PART). Joule links the DOE strategic plan[4] to the underlying base program targets.

[3] http://www.whitehouse.gov/omb
[4] http://www.er.doe.gov/about/Mission_Strategic.htm

Discussion of Q2, Q4 Results

0.4 Results

(SC GG 3.1/2.5.2) 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 SciDAC program. *FY07 efficiency measure:*100%

In FY07, software problems to be benchmarked in OASCR's Joule system have been derived from research in fusion science, chemical combustion, and astrophysics. The primary goal is to demonstrate that each application software effort can be scaled (weakly in the ideal case) on the target machine to execute a meaningful problem instance.

The complexity of a problem instance for the application software is determined according to the number and types of operations, the number and types of operands, and physical resources required for its execution on the target machinery. Experience shows that theoretical estimates of the complexity are extremely important to diagnose issues of efficiency or scalability. Even without a complexity analysis, an image of the program's complexity can be deduced by monitoring the hardware while it executes a program.

`jaguar.ccs.ornl.gov` was the machine utilized to investigate the applications. The fact that the hardware was also being continually upgraded to accomplish the transistion to 250TF scale complicated these exercises but did not change the desired outcome. Some hardware information is provided in the next subsection that attempts to describe the machine as it was benchmarked.

0.4.1 `jaguar.ccs.ornl.gov`

Jaguar[5] is a Cray XT3[6] , XT4[7] installation at ORNL's NCCS leadership class user facility. It is composed of Dual Core AMD64[8] Opteron nodes linked by HyperTransport (HT) to Cray's proprietary SeaStar(2) chips which are used to construct a 3d-torus topology between nodes. The Lustre[9] file system has been implemented with a single Metadata Server (MDS) for 36 Object Storage Servers (OSS) and 144 Object Storage Targets (OST) in three partitions totaling 600TBytes of storage capacity. Each OSS serves two OSTs through two 4Gbit FC (fibre channel)

[5] http://info.nccs.gov/resources/jaguar

[6] http://www.cray.com/downloads/Cray_XT3_Datasheet.pdf

[7] http://www.cray.com/downloads/Cray_XT4_Datasheet.pdf

[8] see, for instance, *AMD64 Technology: AMD64 Architecture Programmer's Manual Volume 1: Application Programming* (http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/24592.pdf)

[9] http://wiki.lustre.org/

cards. There are two nine disk RAID5 groups (called *tiers*) per logical unit number (LUN) -thus 18 disks per LUN. The disks are 10K RPM FC drives with 300GBytes of storage per. The aggregate file system bandwidth is 72GB/s. In all, there are 11,508 compute nodes and 200 io, service nodes. The service nodes are dual core AMD64 Opteron processors with 8GB memory and clock frequency 2.6GHz. Table [1] is a specification reference for some select system features of the compute nodes. The operating system is UNICOS/lc -a combination of SUSE Linux on service nodes and the Catamount microkernel on the compute nodes.

Dual Core AMD-64 Opteron	XT3	XT4	
processor clock frequency	2.6E9 cycle / s	2.6E9 cycle / s	
sockets	5212	6296	11508
processor cores / socket	2	2	
BW / socket	2.1 GB / s	4.1 GB / s	
L1 cache / core	64K Inst, 64K Data	64K Inst, 64K Data	2-way assoc
L2 cache /core	1MB	1MB	16-way assoc
memory / socket	4 GB	4 GB	$>$ 46TB
memory BW	6.4 GB / s	10.656 GB / s	
interconnect	HT - Cray SeaStar	HT - Cray SeaStar2	3d Torus
interconnect BW	7.6 GB / s / port	7.6 GB / s / port	6 ports $\sim 46GB/s$

Table 1: Some specifications of `jaguar.ccs.ornl.gov`. Memory bandwidth can be approximated by the product of the memory frequency and the memory bus width. For the XT3, it is $400E6\frac{cycles}{second} \times 128\frac{bit}{cycle} \times 1\frac{Byte}{8bits} \sim 6.4\frac{GByte}{second}$. For the XT4, it is $666E6\frac{cycle}{second} \times 16\frac{Byte}{cycle} \sim 10.656\frac{GByte}{second}$.

Hardware events

Often PAPI[10] is used to capture hardware events during the execution of a process. Unfortunately, preset definitions for counts of cycles, instructions, data and instruction accesses, and TLB data and instruction accesses are inconsistent in how they are composed on different chip sets and are indeed not meaningful in some cases. The two events that are supported on all platforms are PAPI_TOT_CYC and PAPI_TOT_INS. With a knowledge of the clock frequency of the processor and the cycle count, the total time a process was executed is easily calculated. When collected during a parallel run, the largest time calculated in the hardware allocation is the execution time. On the AMD Opteron, CPU_CLK_UNHALTED is the hardware event to directly access to obtain the measure of cycles. The cycle count is composed by aggregating the following Opteron hardware events: BUSY, IC_FETCH_STALL, FP_NONE_RET. There is insight to be gained with a detailed trace of stalls and it is possible to achieve a more complete breakdown of them, but this analysis will not be pursued here. On the AMD Opteron, the x86 instructions are composed of branch, integer, floating point, and memory instructions. A measure of these can be obtained by direct access of FR_X86_INS. Indeed, overcounting instructions occurs, for example, when speculative instructions are issued but do not complete. On the Opteron, floating point instructions are composed as FP_ADD_PIPE + FP_MULT_PIPE + FP_ST_PIPE + FP_FAST_FLAG. Additionally, FR_FPU_MMX_3D, FR_FPU_SSE_SSE2_PACKED, FR_FPU_SSE_SSE2_SCALAR are SIMD instructions that are not simply classified but need to be included if performed. Overcounting floating point operations can be a problem when processors include floating point stores or

[10] http://icl.cs.utk.edu/papi/

loads in the count or include instructions executed by notional FPUs. Fused operations also pose problems.

Floating Point Performance

The support for streaming SIMD extensions to the AMD Opteron instruction set defines 128-bit media arithmetic instructions that greatly increase performance. Parallel operations exist for vectors composed of words, double words, or quad words. A common example is the support of instructions that operate on vectors composed of two 64-bit double precision numbers. The AMD Opteron can compute 2 floating point operations per cycle per core with SSE2 instructions. Thus, the peak floating point performance of `jaguar.ccs.ornl.gov` *compute nodes* can be easily calculated for a given allocation of processor cores. Considering only compute nodes: $23016 cores \times 2 \frac{flop}{cycle \cdot core} \times 2.6E9 \frac{cycle}{second} \sim 119.6832E12 \frac{flop}{second}$[11]

0.4.2 Chimera

Benchmark Problem

- Q2
  - Post-bounce evolution of 11 solar mass progenitor s11.2
  - 256 radial zones $\times$ 256 angular zones
  - 20 groups to represent the neutrino spectrum
  - alpha network for nuclear burning -14 species
  - Newtonian gravity
  - 10 full transport timesteps
    - hydrodynamic subcycling, 20 subcycles per transport step
  - ray-by-ray-plus approximation -lateral transport between rays is neglected
- Q4
  - 256 radial zones $\times$ 32$\times$64 angular zones
  - same as Q2 otherwise

File i/o was suppressed during the Joule benchmarks of Chimera.

Metric

The observable that Chimera code developers use to relate the performance of their application to the performance of the computer chosen to execute their application is the *measured-wall-time to compute a single subcycled-hydrodynamic-timestep*.

[11] The Basic Linear Algebra Subprograms (BLAS) are categorized according to the complexity of the operation being performed and the number of indices required to probe the structures holding operands. Level-1 BLAS (BLAS-1) perform $O(n)$ operations on $O(n)$ operands. Level-2 BLAS perform $O(n^2)$ operations on $O(n^2)$ operands. Level-3 BLAS perform $O(n^3)$ operations on $O(n^2)$ operands. The functions supported by the BLAS have long formed the backbone of scientific computing software efforts. An example of a BLAS-1 operation is $y \leftarrow \alpha x + y$. If α, x, y are double precision operands, then the operation requires 24 bytes of data to execute 2 floating point operations (αx and $(\alpha x) + y$) or perhaps one fma instruction. Suppose the former. The XT3 has 6.4GB/s bandwidth. Naively, to sustain this operation at peak floating point processor performance in a socket would require $\sim \frac{3 operands}{2 flop} \times 8 \frac{Byte}{operand} \times 2 \frac{flop}{cycle \cdot core} \times 2 core \times 2.6e9 \frac{cycle}{second} = 124.8 \frac{GByte}{second}$. It is challenging to devise methods to achieve sustained performance for such operations that deliver so little data reuse. REF: *An Extended Set of Fortran Basic Linear Algebra Subprograms*, J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson, ACM Transactions on Mathematical Software 14(1): 117, (1988) ; *A Proposal for Standard Linear Algebra Subprograms*, R. Hanson, F. Krogh, and C. Lawson, ACM SIGNUM Newsl. 8 (1973), 16

	Q2	Q4
socket type	XT3	XT3
sockets	128	1024
cores / socket	2	2
wall time [s]	612.12	649.71
PAPI_TOT_INS	nr	2894123008000000
PAPI_FP_INS	13936153320759	nr
<FP_ADD_PIPE>	nr	7.42888e10
<FP_MULT_PIPE>	nr	8.007889e10
<FR_FPU_SSE_SSE2_PACKED>	nr	8.72586e10

Table 2: Some hardware events (machine observables) measured during the Q2 and Q4 production runs of Chimera's Joule problem.

Derived Observations

The scale factor for weak scaling is $2048/256 = 8$.

In both Q2 and Q4, 10 [transport steps] * 20 [subcycled hydrodynamic steps / transport step] = 200 [subcycled hydrodynamic steps] were made.

The Chimera metric in Q2 : wall time / subcycled hydrodynamic step = 612.12[s] / 200[s-h-s] = 3.0606 [s/s-h-s]. The Chimera metric in Q4 : wall time / subcycled hydrodynamic step = 649.71[s] / 200[s-h-s] = 3.24855 [s/s-h-s]. The ratio is : 3.0606 / 3.24855 $\sim$ 0.94214. This number is ideally 1.

The floating point performance of the Q2 benchmark: 13936153320759 [flop] / 612.12 [s] $\sim$ 2.2767028231e10 [FLOPs]. The peak floating point performance for the Q2 machine allocation: 256 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 133.12e10 [FLOPs]. Thus, the Q2 benchmark achieved 1.71% of theoretical peak.

The floating point performance of the Q4 benchmark is constructed from the average hardware event information captured during execution. Total floating point operations executed can be approximated by [<FP_ADD_PIPE>:= 7.42888e10 + <FP_MULT_PIPE>:= 8.007889e10] [flop / core] * 2048 [core] $\sim$ 3.1614502912e14 [flop]. The achieved floating point performance in Q4: 3.1614502912e14 [flop] / 649.71 [s] $\sim$ 4.86594063689e11 [FLOPs]. The peak floating point performance for the Q4 machine allocation: 2048 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 1.06496e13 [FLOPs]. The Q4 benchmark achieved 4.57% of theoretical peak. Another observation about the Q4 run is that $\sim$ 2048 [core] * 8.72586e10 [SSE2 flop / core] / 3.1614502912e14 [flop] $\sim$ 56.53% of the floating point operations were performed with the streaming SIMD extensions to the Opteron instruction set.

Now, in Q2 13936153320759 [flop] were computed. In Q4, one would expect no less than 8 times this for the upscaled problem. Indeed, 3.1614502912e14 [Q4 flop] $>$ 13936153320759 [Q2 flop] * 8 :=1.11489226566072e14 [flop]. The difference in work but similarity in wall times is believable owing to the increased rate of floating point computation in Q4.

Result

Better than weak scaling was achieved for Chimera from the hardware perspective since more than factor of $\frac{2048}{256} = 8$ times floating point operations were performed in normalized measured time. In the new 3D code, two of three spatial dimensions are distributed over $M \times N$ processes, while the third dimension is computed locally. M MPI subcommunicators are first used to partition the radial component into M domains. The domain of the polar coordinate is local. The PPMLR update can be computed with no communication. After transposing back to original

layout, N MPI subcommunicators partition the radial component and the azimuthal coordinate is local. The net effect is two subsequent, but small data transposes instead of one large transpose per fractional timestep.

The new 3D modeling capability reveals new degrees of freedom. In particular the SASI $m = 1$ mode (forbidden in 2D) dominates rather than the $l = 1$ mode. The physical interpretation is that a large amount of angular momentum is delivered onto the proto-neutron star causing it to spin up to periods of $\sim 50ms$ -consistent with observed periods of young pulsars. The new physics establishes that accurate models of supernovae must not only include all the known physics but also be evolved in three dimensions.

0.4.3 GTC-S

Benchmark Problem

- Q2
  - microturbulence particle-in-cell simulation of DIIID tokamak experimental shot 122338
  - temperature and density profiles are 1.6s after initial discharge
  - 100 physical timesteps
  - 64 toroidal planes
  - 78,811 grid points per toroidal plane
  - 1 processor core per toroidal plane; 10 particles per cell
  - 50439040 particles in the volume
- Q4
  - 64 processor cores per toroidal plane; 640 particles per cell
  - 3228098560 particles in the volume
  - same as Q2 otherwise

Metric

The observable that GTC-S code developers use to relate the performance of their application to the performance of the computer chosen to execute their application is the *number-of-particles * timesteps / wall-clock-time*. It is noted the the number of grid points per poloidal plane is determined by size of the fusion device and the physics driving the device.

Derived Observations

The scale factor is $4096/64 = 64$.

The GTC-S metric computed on 64 processor cores in Q2: 50439040 [particles] * 100 [timesteps] / 549.63 [seconds] $\sim$ 9.176908e6 [particles / second / timestep]. The GTC-S metric computed on 64 processor cores in Q4: 50439040 [particles] * 100 [timesteps] / 471.01 [seconds] $\sim$ 1.07087e7 [particles / second / timestep]. The GTC-S metric computed on 4096 processor cores in Q4: 3228098560 [particles] * 100 [timesteps] / 490.12 [seconds] $\sim$ 6.58634e8 [particles / second / timestep]. It is expected that the GTC-S metric will scale weakly. Thus, it is expected that 64 times the number of particles per second per timestep evolved on 64 processor cores can be evolved on 4096 processor cores. 64 * 1.07087e7 [particles / second / timestep] $\sim$ 68.5356e7 [particles / second / timestep]. Comparing this value to the 4096 core run gives 65.8634e7 / 68.5356e7 $\sim$ 96.1% of ideal weak scaling! (same number you get by taking the

	Q2	Q4(1)	Q4(2)
socket type	XT4	XT4	XT4
sockets	32	32	2048
cores / socket	2	2	2
wall time [s]	549.63	471.01	490.12
PAPI_TOT_INS	64192817688103	60210415081552	4097954637879958
PAPI_FP_INS	11576407397891	13691160459066	876286259955885

Table 3: Some hardware events (machine observables) measured during the Q2 and Q4 production runs.

ratio of the runtimes) Also, the agreement in the total instructions and fraction of total floating point operations scales remarkably close to the ideal.

The Q4 64 processor core run outperforms the Q2 64 processor run due to enhancements made to the software during Q3 and Q4. The Q4 64 processor core was used in the above analysis.

The floating point performance of the Q2 benchmark: 11576407397891 [flop] / 549.63 [s] $\sim$ 2.1062182555e10[FLOPs]. The peak floating point performance for the Q2 machine allocation: 64 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 33.280e10 [FLOPs]. Thus, the Q2 benchmark achieved 6.33% of theoretical peak.

The floating point performance of the 64 processor core Q4 benchmark: 13691160459066 [flop] / 471.01 [s] $\sim$ 2.9067664081e10 [FLOPs]. The peak floating point performance for the 64 processor core machine allocation: 64 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 33.280e10 [FLOPs]. Thus, the Q4 64 processor core benchmark achieved 8.73% of theoretical peak.

The floating point performance of the 4096 processor core Q4 benchmark: 876286259955885 [flop] / 490.12 [s] $\sim$ 1.787901452615e12 [FLOPs]. The peak floating point performance for the Q4 machine allocation: 4096 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 2.129920e13 [FLOPs]. Thus, the 4096 processor core Q4 benchmark achieved 8.39% of theoretical peak.

Result

GTC-S clearly scales weakly given the problem and machine parameters tested in FY07. For fixed fusion devices, adding particles is required to evolve the system over more timesteps while keeping the finite particle fluctuations to a low level.

GTC-S employs a generalized model with the realism of tokamak experiments into nonlinear gyrokinetic simulations of plasma turbulence. These include a systematic treatment of plasma rotation and equilibrium $\mathbf{E} \times \mathbf{B}$ flow, realistic plasma profiles and corresponding MHD equilibria. The general geometry simulation capability has been developed with the following favorable features: i) By rescaling the radial coordinate, the grid size in the perpendicular direction is correlated with the local gyroradius which, varying substantially from the core to the edge, defines the spatial scale of turbulence at different locations. ii) Gyrokinetic transformations of potential and charge density between particle and guiding center positions are calculated with a finite ratio (B_θ/B) correction which is a significant geometry effect on the turbulence calculation, particularly for spherical torus devices. iii) The applied equilibrium $\mathbf{E} \times \mathbf{B}$ flow with the spatial scale of the plasma minor radius, which is believed to play an important role in determining the turbulence level, is calculated from a first-principles based particle simulation of global neoclassical dynamics with important finite orbit effects. Working with a symmetry coor-

dinate system, a relatively regular mesh in real space for strongly shaped toroidal plasmas can be constructed. This also facilitates straightforward visualization. In the large aspect ratio circular concentric geometry limit, cross benchmarks of the linear and nonlinear characteristics, such as real frequency, growth rate, steady-state heat flux and zonal flow amplitude, of ITG turbulence have been carried out to validate the general geometry model and simulation.

GTC-S will be used to study electron temperature gradient (ETG) turbulence for NSTX at PPPL. An experimental campaign is now underway to measure the radial spectrum of ETG turbulence. The problem has been studied with GTC with circular geometry finding that the ETG transport is too small to explain the experimental observation from NSTX.

0.4.4 S3D

Benchmark Problem

- Q2
  - slot burner bunsen flame
  - premixed methane-air chemistry, 13 chemical species
  - 17 degrees of freedom / grid point
  - 195 million grid points
  - timesteps
    - XT3, 4500
    - XT4, 3000
- Q4
  - temporally evolving, planar slot-jet with a central fuel slab surrounded by an oxidizer
  - non-premixed ethylene-air chemistry, 19 chemical species
  - 23 degrees of freedom / grid point
  - 341 million grid points
  - 2000 timesteps

Metric

The observable that S3D code developers use to relate the performance of their application to the performance of the computer chosen to execute their application is the *core-hours per grid-point per time-step*.

Derived Observations

The scale factor for weak scaling is $14112/7200 = 1.96$.

The wall times measured in Q2, Q4 are clearly different and therefore need to be normalized according to some problem specific parameters. The changes from Q2 to Q4 are more complex chemistry, more grid points for length scale resolution, and less physical timesteps.

The S3D metric in Q2 on the XT3: 10242.14 [s] / 3600 [s / h] * 7200 [cores] / 195e6 [gp] / 4500 [ts] $\sim$ 2.33439e-8 [core-hours] / [grid point] / [physical timestep] . The S3D metric in Q2 on the XT4: 5569.68 [s] / 3600 [s / h] * 7200 [cores] / 195e6 [gp] / 3000 [ts] $\sim$ 1.90416e-8 [core-hours] / [grid point] / [physical timestep] . The S3D metric in Q4 on the XT3 $\cup$ XT4: 7821.17 [s] / 3600 [s / h] * 14112 [cores] / 341e6 [gp] / 2000 [ts] $\sim$ 4.49545e-8 [core-hours] /

	Q2(1)	Q2(2)	Q4
socket type	XT3	XT4	XT3 $\cup$ XT4
sockets	3600	3600	7056
cores / socket	2	2	2
wall time [s]	10242.14	5569.68	7821.17
PAPI_TOT_INS	155630385570322016	nr	nr
PAPI_FP_INS	38734703861165216	25823255353913720	nr
<PAPI_TOT_INS>	nr	nr	15264495927954
<PAPI_FP_INS>	nr	nr	5651281911147

Table 4: Some hardware events (machine observables) measured during the Q2 and Q4 production runs of S3D's Joule problem.

[grid point] / [physical timestep]. The metric is $O(1.e-8)$ in each case but costs more in Q4 than in Q2.

The total number of floating point operations per timestep in Q2 on the XT3: 38734703861165216 [flop] / 4500 [timestep] $\sim$ 8607711969147.826 [flop/timestep]. The total number of floating point operations per timestep in Q2 on the XT4: 25823255353913720 [flop] / 3000 [timestep] $\sim$ 8607751784637.907 [flop/timestep]. (note : 8607711969148 / 8607751784638 $\sim$ 99.999537% likeness) The total number of floating point operations per timestep in Q4: 5651281911147 [average flop / core] * 14112 [core] / 2000 [timestep] $\sim$ 39875445165053.232 [flop/timestep]. Thus, on average, the Q4 run executes a factor of $\sim$ 4.63 (as opposed to 1.96) more floating point operations each timestep.

The total number of degrees of freedom in the Q2 runs: 17 [degrees of freedom / grid point] * 195e6 [grid points] = 3315000000 [dof]. The total degrees of freedom per compute core in Q2: 3315000000 [dof] / 7200 [core] $\sim$ 460416.67 [dof / core]. The total number of degrees of freedom in the Q4 runs: 23 [degrees of freedom / grid point] * 341e6 [grid points] = 7843000000 [dof]. The total degrees of freedom per compute core in Q4: 7843000000 [dof] / 14112 [core] $\sim$ 555768.141 [dof / core]. Thus, a factor of $\sim$ 1.21 more degrees of freedom are executed per core in Q4 even though there are $\sim$ 1.15 more grid points per core in Q2. This is due to the increase in chemical complexity of the slot-jet configuration problem.

The compute time per degree of freedom per timestep per core on the Q2 XT3 run: 10242.14 [s] / 3315000000 [dof] / 7200 [core] / 4500 [timesteps] $\sim$ 9.5359e-14 [s / dof / timestep / core]. The compute time per degree of freedom per timestep per core on the Q2 XT4 run: 5569.68 [s] / 3315000000 [dof] / 7200 [core] / 4500 [timesteps] $\sim$ 7.7784e-14 [s / dof / timestep / core]. The compute time per degree of freedom per timestep per core for the Q4 run: 7821.17 [s] / 7843000000 [dof] / 14112 [core] / 2000 [timesteps] $\sim$ 3.5332e-14 [s / dof / timestep / core].

The floating point performance of the Q2 benchmark on the XT3: 38734703861165216 [flop] / 10242.14 [s] $\sim$ 3.782e12 [FLOPs]. The floating point performance of the Q2 benchmark on the XT4: 25823255353913720 [flop] / 5569.68 [s] $\sim$ 4.636e12 [FLOPs]. The peak floating point performance for the Q2 machine allocation: 7200 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 37.440e12 [FLOPs]. Thus, the Q2 benchmark achieved 10.10%, 12.38% of theoretical floating point peak on the XT3 and XT4 allocations respectively.

The floating point performance of the Q4 benchmark: 5651281911147 [average flop / core] * 14112 [core] / 7821.17 [s] $\sim$ 10.197e12 [FLOPs]. The peak floating point performance for the Q4 machine allocation: 14112 [cores] * 2 [flop / cycle / core] * 2.6e9 [cycle / second] $\sim$ 73.3824e12[FLOPs]. Thus, the Q4 benchmark achieved 13.89% of theoretical floating point

peak.

A feature that complicates the interpretation of measured S3D results is the hybrid nature of the hardware allocation in Q4 since the 14112 core run cannot be accomodated with only one or the other node type. In Q2, homogeneous allocations of XT3 and XT4 nodes executed the premixed methane-air bunsen slot problem. The difference in measured wall time is a factor of 1.84 despite the fact that the problem and thus the work load were the same. The XT4 injecion bandwidth and cpu-memory bus advantage account for the difference.

Result

The scaling behavior of the S3D software can be observed along different trajectories:

1. increased grid size allowing the simulation of higher Reynolds number regimes by capturing a wider range of length scales
2. more complex chemical composition allowing the simulation of more complex and realistic hydrocarbon fuels
3. longer integration times allowing a more complete temporal development of the solution and larger sample sets for data analysis

It seems that *compute time per degree of freedom per timestep per core* is a good metric for S3D. In Q4, S3D evolved a factor of ~ 2.37 more degrees of freedom on a factor of ~ 1.96 more processor cores per timestep. From a purely hardware event perspective, from Q2 to Q4 S3D was enhanced for performance.

S3D demonstrated the ability to use as much of the machine as necessary to pursue two significant physical problems in FY07.

A significant result of the Q2 run was the confirmation that turbulent eddies can penetrate the preheat zone and increase the mixing process in the TRZ regime. Experimental studies in the TRZ regime have been contradictory as to whether flame thickness relative to laminar flames in the regime increase or decrease. The simulation found that on the Eulerian 3D grid, flame thickness increases relative to laminar flames for moderate ranges of turbulence intensity, however negligible flame thickness increases are observed at high turbulence intensities. The new question is why does flame thickening not increase at the highest intensities. Tracer particles will be implemented to further pursue this and other significant issues for the premixed fuel scenario.

In Q4, S3D achieved its goal of enacting an extinction and reignition event in turbulent non-premixed flames. The role of the scalar dissipation rate due to turbulent fluctuations in the flow field can be studied with the data from this run to obtain insights into the mechanisms that govern and contribute to the reignition process.

Chimera Discussion

0.5 Objective and Expected Significance

The death of massive stars ($M >$ 8-10 $\mathrm{M}_{\odot}$) in core collapse supernovae are an important link in our chain of origin from the Big Bang to the present. They are the dominant source of elements in the periodic table between oxygen and iron [157, 145] and there is growing evidence they are indeed responsible for producing half the elements heavier than iron [4]. Core collapse supernovae serve both to disperse elements synthesized in massive stars during their lifetimes and to synthesize and disperse new elements. Moreover, they are the most energetic explosions in the Universe, and there is now an indisputable connection between peculiar "Type Ic" core collapse supernovae, or hypernovae, and gamma-ray bursts [100, 63]. Both phenomena occur under a common umbrella of massive stellar core collapse and explosion. Moreover, a first-principles understanding of long, soft gamma-ray bursts must begin with an understanding of stellar core collapse and the physics involved in core collapse supernovae for both "ordinary" (Type II, Ib, ordinary Ic) supernovae and hypernovae. Thus, our proposed work can also be thought of as the first stage in a longer-term effort to simulate, from progenitor to burst, hypernovae and long, soft gamma-ray bursts.

As the name suggests, core collapse supernovae are initiated by the collapse of the iron cores of massive stars at the end of their lives. The collapse proceeds to ultrahigh densities, in excess of the densities of nucleons in the nucleus of an atom ("super-nuclear" densities). The inner core becomes incompressible under these extremes, bounces, and, acting like a piston, launches a shock wave into the outer stellar core. This shock wave will ultimately propagate through the stellar layers beyond the core and disrupt the star in a core collapse supernova explosion. However, the shock stalls in the outer core, losing energy as it plows through it, and exactly how the shock is revived is unknown. This is the central question in core collapse supernova theory. (For a more complete review, the reader is referred to [104].)

The primary objective of our work is to perform three-dimensional core collapse supernova simulations with most of the known-to-be-important macroscopic physics components, with an eye toward determining the core collapse supernova mechanism and making more accurate predictions of observables associated with such supernovae than were possible with past, two-dimensional models.

0.6 Background and Approach

0.6.1 Laying the Foundation

After core bounce, $\sim 10^{53}$ erg of energy in the form of neutrinos and antineutrinos of all three flavors (electron, muon, and tau) is released from the newly formed proto-neutron star (PNS) at the center of the explosion. The supernova explosion energy is $\sim 10^{51}$ erg. Past simulations [151, 14] demonstrate that energy in the form of neutrinos emerging from the PNS can be deposited behind the shock and may revive it. This neutrino reheating is central to core collapse supernova models today. However, while a prodigious amount of neutrino energy emerges from the PNS, the neutrinos are weakly coupled to the material directly below the shock, and, in fact, the neutrino heating is very sensitive to the distribution of neutrinos in energy (or frequency) and direction of propagation, at any given spatial point behind the shock [32, 79, 102, 106, 107, 77]. In turn, this ultimately requires "multiangle," "multifrequency" (Boltzmann) neutrino transport in order to compute accurately the neutrino distributions in this region. This renders the core collapse supernova problem a truly multidimensional (space plus neutrino angles and energy), petascale (petaflops, petabytes) problem.

The neutrino heating may be aided by fluid instabilities (*e.g.*, convection) in the PNS [142, 154, 111, 24, 30], which may boost the luminosity of this central neutrino bulb. Convection directly beneath the shock fundamentally alters the nature of neutrino shock reheating [69, 33, 79, 60, 30, 27] relative to the spherically symmetric case, allowing simultaneous downflows that fuel the neutrino luminosities and upflows that bring energy to the shock. And the newly discovered instability of the shock wave itself, the Stationary Accretion Shock Instability (SASI), will likely dramatically alter the shock and explosion dynamics [15, 78, 35, 123]. The centrifugal effects of stellar core rotation [60, 30], and other of its effects, will also change supernova dynamics qualitatively and quantitatively, and stellar core magnetic fields, increased perhaps dramatically by compression during collapse, convection (*e.g.*, via a dynamo), and rotation (through wrapping and shear; in the latter case the magnetorotational instability may occur and, if so, would dominate the field evolution), may also play a significant role in driving, and perhaps collimating, core collapse supernova explosions [144, 1, 31].

Finally, the nuclear abundances should be evolved in regions where nuclear statistical equilibrium (NSE) cannot be maintained. This will enable the potentially observable products of nucleosynthesis to be followed and, most important for the mechanism, the energy released by nuclear burning to be fed back into the computation of the explosion dynamics. While the energy released is rather small in overall amount, we will describe below that this energy release occurs in critical regions and has a decided influence on the dynamics if all other factors give rise to a very marginal outcome.

While the list of major macroscopic components in any core collapse supernova clearly indicates this is a three-dimensional phenomenon, three-dimensional studies performed by this investigator team in the past with a reduced set of physics (focused on certain critical aspects of the supernova problem) demonstrate how different the outcomes will likely be in more complete three-dimensional models. In particular, simulations in two spatial dimensions that led to the discovery of the SASI [15] showed

the dominance of the $l = 1$ SASI mode in this instability and in defining the resultant explosion dynamics and morphology. On the other hand, simulations performed now in three dimensions demonstrate that the $m = 1$, not the $l = 1$, mode dominates [17, 19, 20]. The $m = 1$ mode is not admitted in the two-dimensional models given that it violates axisymmetry. Thus, three-dimensional models allow entirely new degrees of freedom, uncover the existence of new SASI modes, and, in turn, the new SASI modes led to yet new surprises: the startling discovery that the SASI can give rise to a spiraling flow beneath the supernova shock wave capable of depositing significant angular momentum onto the PNS, causing it to spin up. The predicted spin periods are ~ 50 ms, in the range of the observed spin periods of young pulsars [17, 16, 19]. Moreover, such spins can be generated beginning with nonrotating progenitors, complicating the link between progenitor characteristics and final neutron star properties. These simulations provide specific evidence that core collapse supernova models must be performed in three dimensions. Thus, three-dimensional simulations will open a new window on supernova modeling and the supernova mechanism. Such simulations — i.e., three-dimensional multi-physics simulations with significantly increased realism in the model components — were not possible before.

Indeed, current 3D models have yet to include all of the following: (1) multifrequency, or multiangle and multifrequency, neutrino transport, (2) convection and other fluid instabilities, (3) rotation, (4) nuclear burning, and (5) magnetic fields [60, 17, 16, 155]. Only (2) and (3) have been included in models without neutrino transport or with gray (frequency-integrated) neutrino transport. *Our primary scientific goal will be to perform 3D core collapse supernova simulations that include (1)–(4) to ascertain their explosion mechanism and to predict their associated observables.*

0.6.2 A Staged Approach

Three-dimensional simulations with multiangle, multifrequency (Boltzmann) neutrino transport at sufficient resolution are not currently possible. They will require multi-petaflop computing platforms, which will not be available until 2010 or later. However, current supercomputers will permit significant advances to be made in three-dimensional supernova modeling through a key intermediate step: the performance of three-dimensional simulations with multifrequency (but not multiangle) neutrino transport. Moreover, the inclusion of multifrequency neutrino transport in three-dimensional models can be advanced in two stages: (1) multifrequency neutrino transport in the "ray-by-ray-plus" approximation [130] and (2) three-dimensional multifrequency neutrino transport. The work proposed here falls under stage (1). In the ray-by-ray-plus approximation, neutrino transport is followed along radial rays, but lateral transport between rays is neglected. Essentially, the compromise in stage (1) is to reduce the spatial dimensionality of the transport in order to include the extra dimension of neutrino energy (frequency). As described above, the neutrino shock reheating is very sensitive to the neutrino spectra. Consequently, the inclusion of multifrequency neutrino transport is the primary step in moving to three spatial dimensions. Moreover, stage (1) allows for existing neutrino transport codes to be used in performing the three-dimensional simulations, whereas stage (2) requires extensive new code devel-

opment. The approach in stage (1) has also been adopted by the Max Planck group in performing their two-dimensional simulations.

0.7 CHIMERA Overview

CHIMERA can well be described as a "chimera" of three, separate, rather mature codes. The codes are tightly coupled in a single executable through a set of interface routines that provide each of the code components access to global variables for use. The primary code modules are designed to evolve the stellar gas hydrodynamics (VH1), the "ray-by-ray-plus" neutrino transport (MGFLD-TRANS), and the thermonuclear kinetics (XNET). These three "heads" are augmented by a sophisticated equation of state (EOS) for nuclear matter (e.g. LS-EOS [91]) (The EOS provides a relationship between the matter density and temperature and the resulting pressure, closing the system of radiation hydrodynamics equations.) and a self-gravity solver capable of an approximation to general-relativistic gravity. The hydrodynamics is directionally split, and the ray-by-ray transport and the thermonuclear kinetics solve occur "during" the radial sweep, when all the necessary data for those modules is local to a processor (see Figure 24). The individual modules are algorithmically coupled in an operator split approach. This approach is well motivated, as the characteristic time scales for each module are widely disparate. Specifically, during the radial sweep of the hydrodynamics, the neutrino transport and the thermonuclear burning are computed along each radial ray, using only data that is local to that ray and, therefore, local to the current process.

The hydrodynamics module in CHIMERA is a modified version of the PPM code VH-1, which has been widely used in astrophysical fluid dynamics simulations and as an important benchmark code for a variety of platforms. VH-1 is a Lagrangian remap implementation of the Piecewise Parabolic Method (PPM) [44]. Being third order in space (for equal zoning) and second order in time, the code is well suited for resolving shocks, composition discontinuities, etc. with modest grid requirements. To avoid the odd-even decoupling and carbuncle phenomenon for shocks aligned parallel to a coordinate axis we have employed the local oscillation filter method of Sutherland et al. (2003) which subjects only a minimal amount of the computational domain to additional diffusion. Redshift and time dilation corrections are included in both the hydrodynamics and neutrino transport (both are consequences of general relativity). A moving radial grid option, where the radial grid follows the average radial motion of the fluid, makes it possible for the core infall phase to be followed with good resolution. We plan to develop a PPM-based magnetohydrodynamics capability as well, enabling the inclusion of stellar magnetic fields in CHIMERA simulations. This development will take considerable effort, and may alter the computational intensity of the hydrodynamics module. Nevertheless, we propose to undertake this addition, as it would enable inclusion of *all* known important physical processes in CHIMERA supernova simulations.

The equation of state (EOS) of [91] is currently employed for matter at high densities. At densities below about $1.7 \times 108 \frac{g}{cm^3}$ the state of the matter is described by mixture of 4 nuclei (neutrons, protons, helium, and a representative heavy nu-

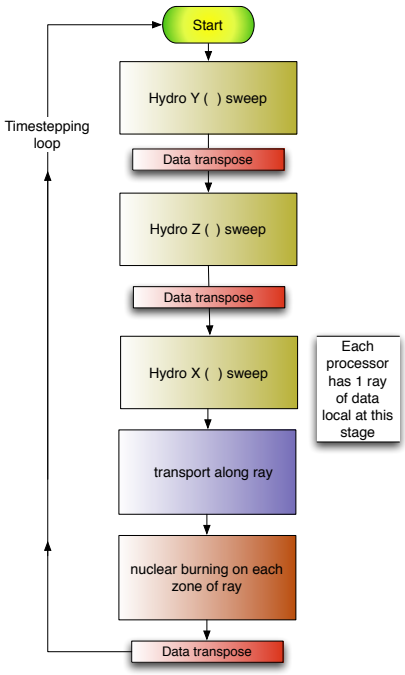


Figure 1: Schematic CHIMERA flowchart.

cleus) in a highly modified version of the EOS described by [45]. For regions not in NSE (at even lower densities), an EOS with a nuclear component consisting of 14 α-particle nuclei (^{4}He to ^{60}Zn), protons, neutrons, and an ironlike nucleus is used. An electron-positron EOS with arbitrary degeneracy and degree of relativity spans the entire density-temperature regime of interest. The nuclear composition in the non-NSE regions is evolved by the thermonuclear reaction network of [73]. The thermonuclear network is a set of coupled ODEs describing the various nuclear processes that transmute the isotopic composition of the matter. This is a fully implicit general purpose reaction network; currently, we have used a smaller nuclear dataset to implement a so-called α-network, i.e. only reactions linking the 14 α nuclei from ^{4}He to ^{60}Zn are used. Data for these reactions is drawn from the REACLIB compilations [134], the world standard for reaction data in simulations of this type. Detailed nucleosynthesis requires evolving 150 or more nuclear isotopes throughout the matter which is ultimately ejected. Simple replacement of the 14 element network with 150 isotopes in the fully implicit Backward Euler integration scheme represents a several hundred-fold increase in the computational cost. Given the current $\approx$5% of execution time required by the alpha network in the current simulations, this cost is prohibitive if implemented naively. However, we can achieve computational savings of roughly a factor of 10 by employing techniques using local partial equilibria (termed quasi-equilibrium or QSE) to reduce size of the system of equations which must be integrated [71].

Ideally, neutrino transport would be implemented with full multidimensional Boltzmann transport. This important development, however, lies beyond even the petascale, and may well require exascale computing to fully realize. We compromise by implementing a "ray-by-ray-plus" approximation [30, 29] for neutrino transport, whereby the lateral effects of neutrinos such as lateral pressure gradients (in optically thick conditions), neutrino advection, and velocity corrections are taken into account, but transport is performed only in the radial direction. This approximation offers us the ability to include the most up-to-date microphysics, which is essential to the realistic computation of the crucial coupling of neutrinos to matter in the important region behind the shock. It thus permits us to capture most of the essential physics above the neutrinosphere (the effective radiating surface for neutrinos, analogous to the Sun's photosphere) where the neutrinos are primarily radially directed and where we believe the essential ingredients of the supernova mechanism reside. However, it is well known that this approach tends to overestimate heating in radial rays where hot spots occur near the neutrinosphere, whereas true multidimensional transport will direct some of the neutrino flux from these hot spots in the lateral direction, thereby producing a more uniform spatial neutrino distribution. Some tests [30] show that the effect of the ray-by-ray plus approximation on the temporally averaged neutrino heating is probably only a few percent. We continue to attempt to quantify the impact of the ray-by-ray approximation with our own set of verification tests. Transport is computed by means of multigroup flux-limited diffusion with a flux limiter that has been tuned to reproduce Boltzmann transport results to within a few percent [94]. All $O(v/c)$ observer corrections have been included. The transport solver is fully implicit and solves for four neutrino flavors (types) simultaneously (i.e., ν_e's, $\bar{\nu}_e$'s, ν_μ's and ν_τ's (collectively ν_x's), and $\bar{\nu}_\mu$'s and $\bar{\nu}_\tau$'s (collectively $\bar{\nu}_x$'s)), allowing for neutrino neutrino scattering and pair-exchange, and different neutrino and antineutrino opacities. The PPM technology has been di-

rectly applied to both the spatial and energy advection of neutrinos in both the radial and lateral directions. The neutrino opacities employed for the simulations are the standard ones described in [25], with the isoenergetic scattering of nucleons replaced by the more exact formalism of [136], which includes nucleon blocking, recoil, and relativistic effects, and with the addition of nucleon–nucleon bremsstrahlung [67] with the kernel reduced by a factor of five in accordance with the results of [66]. The transport solver can be replaced with a module for spherically-symmetric Boltzmann neutrino transport (i.e. angular information can be incorporated, enabling the shape of the radiation field at each spatial point to be determined). We have an existing spherically symmetric multigroup Boltzmann transport (MGBT) solver for supernovae [94, 109] that can be straightforwardly adapted as a CHIMERA module. Further testing of the ray-by-ray approximation and our MGFLD solver will be an integral part of our verification testing for CHIMERA, both unit testing of the transport module by comparison to radiation transport test problems and by whole-code testing through comparing the results of CHIMERA simulations incorporating MGFLD to simulations with the new MGBT solver.

0.8 CHIMERA's Hydrodynamics Module

0.8.1 Solving the Hydrodynamics Equations

The hydrodynamics module uses the Piecewise Parabolic Method described by [44] to solve the equations of ideal inviscid compressible fluid flow. These equations, written in conservative, Eulerian form are:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1}$$

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p = \rho \mathbf{a}, \tag{2}$$

$$\partial_t (\rho E) + \nabla \cdot (\rho E \mathbf{u} + p\mathbf{u}) = \dot{\epsilon} + \rho \mathbf{u} \cdot \mathbf{a}, \tag{3}$$

where $\rho, \mathbf{u}, E$ are the mass density, fluid velocity, and total specific energy per unit mass. The total energy is composed of the bulk kinetic energy and the internal energy, $E = u^2/2 + e_m$. The gas pressure, p, is related to the internal energy of the gas through an appropriate equation of state. The source terms for these conservation equations include a local acceleration, $\mathbf{a}$, due to gravity and momentum exchange with the neutrinos, and heating/cooling rate, $\dot{e}psilon$, due to neutrino interactions and nuclear burning. Contributions to these source terms can be found in other sections of this report.

We solve these equations on a spherical polar grid (coordinates r, θ, ϕ), using operator splitting to separate the 2- or 3-dimensional problem into 1-dimensional updates. This 1D update is done in Lagrangian (mass) coordinates followed by a conservative remap back to the original Eulerian grid. Thus the fundamental building block of the hydro module is a one-dimensional update of the Lagrangian fluid equations followed by a conservative remap. The following is an outline of the 1D PPMLR algorithm employed in CHIMERA. For specific details see [44]. Here we consider the equations of ideal hydrodynamics in Lagrangian coordinates in one dimension, with the mass

coordinate m related to the spatial coordinate x by

$$m(x) = \int_{x_o}^{x} \rho(x) x^{\alpha} dx. \tag{4}$$

Here α is 2 for the radial coordinate and 0 for the two angular coordinates (ϕ and $\cos\theta$). Written in conservative form these equations are:

$$\begin{aligned} \partial_t u + x^{\alpha} \partial_m p &= a \\ \partial_t E + \partial_m x^{\alpha} u p &= ua + \dot{\epsilon}, \end{aligned} \tag{5}$$

In the following finite-difference equations we use the subscript j to refer to zone averaged values, subscripts $j-1/2$ and $j+1/2$ refer to values at the left and right hand sides of the zone, and the superscript is the timestep. The spatial coordinate is updated knowing the time-averaged velocity of the zone interface, $\bar{u}_{j+1/2}$:

$$x_{j+1/2}^{n+1} = x_{j+1/2}^{n} + \Delta t \; \bar{u}_{j+1/2}, \tag{6}$$

and the density is updated given the resulting change in volume for a mass element:

$$\rho_j^{n+1} = \frac{\rho_j^n \; Vol_j^n}{Vol_j^{n+1}}. \tag{7}$$

The volume of a zone is given by

$$\begin{aligned} Vol_j &= \frac{1}{3}(r_{j+1/2}^3 - r_{j-1/2}^3) \\ &= (\cos\theta_{j-1/2} - \cos\theta_{j+1/2}) \\ &= (\phi_{j+1/2} - \phi_{j-1/2}) \end{aligned} \tag{8}$$

for updates in the (r, θ, ϕ) direction, respectively.

The momentum and energy conservation equations can be finite differenced as:

$$\begin{aligned} u_j^{n+1} = u_j^n &+ \frac{\Delta t}{\Delta m_j}(\bar{p}_{j-1/2} - \bar{p}_{j+1/2}) \frac{(A_{j+1/2} + A_{j-1/2})}{2} + \frac{\Delta t}{2}(a_j^n + a_j^{n+1}) \\ E_j^{n+1} = E_j^n &+ \frac{\Delta t}{\Delta m_j}(A_{j-1/2}\bar{u}_{j-1/2}\bar{p}_{j-1/2} - A_{j+1/2}\bar{u}_{j+1/2}\bar{p}_{j+1/2}) \\ &+ \frac{\Delta t}{2}(u_j^n a_j^n + u_j^{n+1} a_j^{n+1}), \end{aligned} \tag{9}$$

where the change in the cross-sectional area of a zone is given by

$$\begin{aligned} A_{j+1/2} &= \frac{(r_{j+1/2}^{n+1})^3 - (r_{j+1/2}^n)^3}{3(r_{j+1/2}^{n+1} - r_{j+1/2}^n)} \\ &= \frac{\cos\theta_{j+1/2}^n - \cos\theta_{j+1/2}^{n+1}}{(\theta_{j+1/2}^{n+1} - \theta_{j+1/2}^n)} \\ &= 1 \end{aligned} \tag{10}$$

for the three different directions. Note that the components of momentum transverse to the update direction do not change during the 1D Lagrangian update.

The variables $\bar{u}$ and $\bar{p}$ are the time-averaged values of the velocity and pressure at the (Lagrangian) zone interfaces. The approach of Godunov's method is to obtain these time-averaged quantities by approximating the flow at each zone interface during each timestep with a Riemann shock tube problem. At the beginning of the timestep the zone interface is modelled as a discontinuity separating two uniform states given by the zone averages on the left and right side of the zone interface, e.g, $u_{j+1/2,L}$ is the zone average of the velocity in zone j serving as the left state for the interface at $x_{j+1/2}$ and $u_{j+1/2,R}$ is the zone average of the velocity in zone $j+1$ serving as the right state for the same interface.

The constructed Riemann problem at each zone interface ($j+1/2$) is then given by

$$\begin{aligned} \frac{\bar{p}-p_L}{W_L} + (\bar{u}-u_L) = 0, \qquad & W_L^2 = (\gamma p_L \rho_L)\left(1+\frac{\gamma+1}{2\gamma}(\frac{\bar{p}}{p_L}-1)\right) \\ \frac{\bar{p}-p_L}{W_R} + (\bar{u}-u_R) = 0, \qquad & W_R^2 = (\gamma p_R \rho_R)\left(1+\frac{\gamma+1}{2\gamma}(\frac{\bar{p}}{p_R}-1)\right) \end{aligned} \tag{11}$$

which is solved to find the time-averaged value of the velocity and pressure at this zone interface, $\bar{u}$ and $\bar{p}$. The solution of this Riemann problem typically requires some kind of iterative procedure, as described in van Leer (1979) and in Woodward (1986). CHIMERA uses a Newton iteration with a tolerance of 10^{-4} (i.e., if $\bar{p}$ changes by less than this fraction the iteration stops) and a maximum of 8 iterations. In the absence of shocks $\bar{p}$ is simply the average of p_L and p_R and only one iteration is required.

PPM improves upon this method by using more accurate guesses for the input states to the Riemann problem (the values on either side of the interface). The first step in this approach is to construct a quadratic piecewise continuous interpolation of each fluid variable in each zone. If a is the fluid variable and ξ is the fractional distance within a zone,

$$\xi = \frac{x - x_{j-1/2}}{x_{j+1/2} - x_{j-1/2}} \tag{12}$$

the interpolation process amounts to calculating the 3 coefficients in the parabolic fit given by

$$a(\xi) = a_0 + \xi(\Delta a + a_6(1-\xi)). \tag{13}$$

We follow the algorithm of [44] in calculating these coefficients including the application of 3 separate monotinicity constraints on the interpolation parabolae. In the case of uniform zone spacing and no required monotinicity constraints one arrives at the following definitions:

$$\begin{aligned} a_{0,j} &= \frac{7}{12}(a_j + a_{j+1}) - \frac{1}{12}(a_{j+2} + a_{j-1}), \\ \Delta a_j &= a_{0,j+1} - a_{0,j}, \\ a_{6,j} &= 6(a_j - \frac{1}{2}(a_{0,j} + a_{0,j+1}). \end{aligned} \tag{14}$$

The Riemann input states are taken to be the average of these parabolic interpolations over that part of the zone that can reach the zone interface by a sound wave in a

time Δt, i.e., the characteristic domain of dependence, $x_{j+1/2} \pm c_s \Delta t$. The left and right input states for variable a at zone interface $j+1/2$ are then

$$\begin{aligned} a_L &= a_{0,j+1} - \xi/2\,(\Delta a_j - (1-2\xi/3)a_{6,j}) \\ a_R &= a_{0,j+1} + \xi/2\,(\Delta a_{j+1} + (1-2\xi/3)a_{6,j+1})\,, \end{aligned} \tag{15}$$

where $\xi = c_s \Delta t/\Delta x$. These input states for density, velocity and pressure are then modified to account for local accelerations (both real and ficticious grid forces) before iterating on the Riemann solution to find $\bar{p}$ and $\bar{u}$.

Once the Lagrangian hydrodynamic equations have been differenced to obtain the values of ρ, u, and E at $t + \Delta t$, the conserved fluid variables (mass, momentum, and total energy) can be instantaneously remapped from the Lagrangian coordinate system to the stationary Eulerian grid. This remap step uses the same quadratic interpolation method that was used in the hydrodynamics step, including the use of 'flattening' the parabolic fits in regions of local extrema to suppress numerical oscillations. This remap process begins by constructing new piecewise parabolic interpolations for density, the three velocity components and total energy on the updated Lagrangian grid (x_j^{n+1}). The amount of mass to be remapped from Lagrangian zone j to Eulerian zone $j+1$ (if $\bar{u}_{j+1/2} > 0$) is found by integrating the interpolation function in zone j from $x_{j+1/2}^n$ to $x_{j+1/2}^{n+1}$:

$$dm_{j+1/2} = (\rho_{0,j+1} - \xi/2(\Delta\rho_j - (1-2\xi/3)\rho_{6,j}))\, dV_{j+1/2}, \tag{16}$$

where $\xi = (x_{j+1/2}^{n+1} - x_{j+1/2}^n)/(x_{j+1/2}^{n+1} - x_{j-1/2}^{n+1})$ is the fractional distance the Lagrangian zone interface has moved and

$$\begin{aligned} dV_{j+1/2} &= \left((r_{j+1/2}^{n+1})^3 - (r_{j+1/2}^n)^3\right)/3 \\ &= R(\cos\theta_{j+1/2}^n - \cos\theta_{j+1/2}^{n+1}) \\ &= R(\phi_{j+1/2}^{n+1} - \phi_{j+1/2}^n) \end{aligned} \tag{17}$$

is the volume of the subshell between the updated Lagrangian zone interface and the original Eulerian zone interface for updates in each of the three coordinate directions. Here R is the fixed radius during updates in the θ and ϕ directions.

If, instead, $\bar{u}_{j+1/2} < 0$ then the mass to be remapped from zone $j+1$ to j is

$$dm_{j+1/2} = (\rho_{0,j+1} + \xi/2(\Delta\rho_{j+1} + (1-2\xi/3)\rho_{6,j+1}))\, dV_{j+1/2}, \tag{18}$$

with $\xi = (x_{j+1/2}^n - x_{j+1/2}^{n+1})/(x_{j+1/2}^{n+1} - x_{j-1/2}^{n+1})$. The amount of momentum to be remapped is found by integrating the velocity interpolation and multiplying by the mass in the subshell.

$$\begin{aligned} dmu_{j+1/2} &= dm_{j+1/2}\,(u_{0,j+1} - \xi/2(\Delta u_j - (1-2\xi/3)u_{6,j}))\,, \qquad \bar{u}_{j+1/2} > 0 \\ dmu_{j+1/2} &= dm_{j+1/2}\,(u_{0,j+1} + \xi/2(\Delta u_{j+1} + (1-2\xi/3)u_{6,j+1}))\,, \qquad \bar{u}_{j+1/2} \end{aligned} \tag{19}$$

The final step of the remap process is to find the mass in the evolved Lagrangian zone (which is the same mass as in the unevolved Eulerian zone) and the new mass in the updated Eulerian zone:

$$M_{L,j} = \rho_j^n Vol_j$$

$$M_{E,j} \quad = \quad M_{L,j} + dm_{j-1/2} - dm_{j+1/2}, \tag{20}$$

where Vol_j retains the previous definition for the volume of Eulerian zone j. The velocity on the Eulerian grid is then updated by remapping momentum:

$$u_j^{n+1} = (u_j^n M_{L,j} + dmu_{j+1/2} - dmu_{j-1/2})/M_{E,j}. \tag{21}$$

This same process is done for all 3 components of the moment and for the total energy. In order to improve the conservation of internal energy in situations that should remain strictly isentropic, we instead apply this remap procedure to the internal energy and reconstruct the total energy from its remapped components (rather than constructing internal energy by taking the remapped total energy and subtracting the other components). In the vicinity of a shock, however, the total energy must be remapped in order to arrive at the correct Rankine-Hugoniot shock jump conditions.

In this standard approach of treating gravitational acceleration as an external source term, one finds that the total energy of the star - including self-gravity - is not strictly conserved. We can significantly reduce this problem if instead we redefine the total energy to include the gravitational contribution as well as the kinetic and internal energies. To address similar issues, it has long been standard practice in stellar physics to include the gravitational energy in the evolved "total" energy [93, 42, 87, 51], moving the effects of gravity to the left-hand side of the conservation equation. This is straightforward in these spherically symmetric and frequently Lagrangian formulations, with the conservation equation taking the form

$$\partial_t(\rho(E - GM(r)/r)) + \nabla \cdot (\rho E\mathbf{u} - GM(r)/r + p\mathbf{u}) = 0. \tag{22}$$

where G is Newton's constant and $M(r)$ is the mass enclosed within radius r of the center of the star. This is a homogeneous conservative equation for a globally conserved energy, $E' = E - GM(r)/r$. This re-formulation of the energy equation is made possible by the fact that spherical symmetry makes changes in the gravitational potential "local", due either to radial movement of the Langrangian mass element or mass flow through an Eulerian boundary. This is not true in the general multi-dimensional case.

If, for the general multidimensional case we define a total energy $E' = E + \Phi$, the energy conservation equation becomes,

$$\partial_t(\rho E) + \nabla \cdot (\rho E\mathbf{u} + p\mathbf{u}) = \partial_t\Phi. \tag{23}$$

This leaves only the (hopefully) more slowly varying changes in the global potential to be accounted for as a source term. To implement this approach we do not include gravitational acceleration in the total energy (E') evolution during the 1D PPMLR update, but we do include it in the velocity update. This updated total energy is then conservatively remapped for each directional update, after which the total energy is updated by finite-differencing the source term, $\partial_t\Phi$.

Our implementation of this 3D hydrodynamics module on platforms reaching tens of thousands of processors maintains scalability by keeping the basic unit, the 1D PPMLR update, local to each processor. This is accomplished by using $M \times N$ processors to break up two of the three spatial dimensions, leaving one dimension (lets start

with r) local to each processor. We then use M subcommunicators to transpose the data (using MPI_ALLTOALL) such that the radial coordinate is now sliced up into M domains but the entire domain of the polar coordinate is now local to each processor. At this point the PPMLR update can be done for the θ direction with no communication. After a transpose back to the original data layout, we use N subcommunicators to transpose the data such that the radial coordinate is sliced up into N domains but the azimuthal coordinate is now local.

0.8.2 Solving the Poisson Equation

The use of a spherical grid to facilitate the neutrino transport calculations leads us to the algorithm of Muller and Steinmetz (1995) for solving Poisson's equation in spherical coordinates. Their approach begins with the integral form of Poisson's equation,

$$\Phi(\mathbf{x}) = -G \int \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} d^3\mathbf{x}', \tag{24}$$

and expands the integrand in terms of spherical harmonics, $Y_{lm}(\theta, \phi)$. In the end this approach will have the distinct advantage of requiring only a single global sum across processors. Furthermore, the computational work per zone scales only as $N_r(L+1)^2$, where N_r is the number of zones in the radial direction and L is the order of the highest spherical harmonics used in the expansion of the above integral. As a result, this algorithm is both efficient and highly scalable.

The details of this algorithm are carefully described by Muller and Steinmetz (1995), so we only briefly describe the method here. We begin by expanding the Poisson integral in spherical harmonics:

$$\begin{aligned} \Phi(r,\theta,\phi) &= -G \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^{l} Y_{lm}(\theta,\phi) \int_{4\pi} d\Omega' Y_{lm}(\theta',\phi') \\ &\quad \left(\frac{1}{r^{l+1}} \int_0^r dr'\, r'^{l+2} \rho(r',\theta',\phi') + r^l \int_r^{\infty} dr'\, r'^{1-l} \rho(r',\theta',\phi') \right) \end{aligned} \tag{25}$$

One then need only calculate the local zone integrals

$$\begin{aligned} A_{ijk}^{lm} &= \int_{\phi_{k-1}}^{\phi_k} \int_{\theta_{j-1}}^{\theta_j} \sin\theta \, d\theta \, d\phi \, Y_{lm}^*(\theta,\phi) \int_{r_{i-1}}^{r_i} dr \, r^{l+2} \rho(r,\theta,\phi) \\ B_{ijk}^{lm} &= \int_{\phi_{k-1}}^{\phi_k} \int_{\theta_{j-1}}^{\theta_j} \sin\theta \, d\theta \, d\phi \, Y_{lm}^*(\theta,\phi) \int_{r_{i-1}}^{r_i} dr \, r^{1-l} \rho(r,\theta,\phi), \end{aligned} \tag{26}$$

and sum them over the 3D spherical grid

$$\begin{aligned} C_n^{lm} &= \sum_{i=1}^{n} \sum_{j=1}^{N_\theta} \sum_{k=1}^{N_\phi} A_{i,j,k}^{lm} \\ D_n^{lm} &= \sum_{i=n+1}^{N_r} \sum_{j=1}^{N_\theta} \sum_{k=1}^{N_\phi} B_{i,j,k}^{lm}. \end{aligned} \tag{27}$$

The potential at any grid point (n, o, p) can then be generated by a sum of the spherical harmonics weighted by these coefficients:

$$\Phi_{nop} = -G \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \left[\frac{1}{r_n^{l+1}} \sum_{m=-l}^{l} C_n^{lm} \, Y_{lm}(\theta_o, \phi_p) \right. \\ \left. r_n^l \sum_{m=-l}^{l} D_n^{lm} \, Y_{lm}(\theta_o, \phi_p) \right]. \quad (28)$$

In practice one assumes the density is a slowly varying function of (θ, ϕ), such that the angular zone integrals in eqn. (3) can be pre-computed once assuming a constant density within each zone. The first step in this algorithm is thus to numerically compute, for each j and k, the angular weights

$$a_{jk}^{lm} = \int_{\phi_{k-1}}^{\phi_k} e^{-im\phi} d\phi \int_{\theta_{j-1}}^{\theta_j} P_{lm}(\theta) \, \sin\theta \, d\theta \quad (29)$$

The integral over ϕ is done analytically, but the integral over θ is done numerically. Beer and Podsiadlowski (2002) assume a locally linear density profile in computing the radial integral in eqn. (3), while Muller and Steinmetz (1995) assume a constant density within the zone such that the integral can be solved analytically. We follow the latter approach. In this case

To make this calculation of Φ more efficient we rearrange the order of these summations as suggested by Muller and Steinmetz (1995). We begin by computing the weighted moments for each l and m at each radial zone i:

$$b_i^{lm} = \sum_{j=1}^{N_\theta} \sum_{k=1}^{N_\phi} a_{j,k}^{lm} \rho_{ijk} \quad (30)$$

It is only this summation that requires any global communication given our domain decomposition into 1D radial rays.

We then build the inner and outer sums over radius corresponding to eqn. (4), incorporating the radial integrals of eqn. (3) and the radial coefficients of eqn. (5).

$$\begin{aligned} c_i^{lm} &= c_{i-1}^{lm} \left(\frac{r_{i-1/2}}{r_{i+1/2}} \right)^{l+1} + \frac{b_i^{lm}}{l+3} \left(r_{i+1/2}^2 - r_{i-1/2}^2 \left(\frac{r_{i-1/2}}{r_{i+1/2}} \right)^{l+1} \right) \\ d_{i-1}^{lm} &= d_i^{lm} \left(\frac{r_{i-1/2}}{r_{i+1/2}} \right)^{l} + \frac{b_i^{lm}}{2-l} \left(r_{i+1/2}^2 \left(\frac{r_{i-1/2}}{r_{i+1/2}} \right)^{l} - r_{i-1/2}^2 \right) \end{aligned} \quad (31)$$

Finally, we compute the potential by adding all the moments together.

$$\Phi_{nop} = -G \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} Y_{lm}(\theta_o, \phi_p)(c_n^{lm} + d_{n-1}^{lm}) \quad (32)$$

where the sum over $m < 0$ is accounted for by doubling the corresponding positive term and dropping the imaginary component.

Both authors report maximum errors of order 1% in the computed potential of an oblate spheroid, and find no significant improvement beyond $l = 14$.

0.9 CHIMERA's Transport Module

0.9.1 Overview

Neutrino transport is implementing in CHIMERA by means of a "ray-by-ray-plus" approximation [30] for neutrino transport whereby the lateral effects of neutrinos such as lateral pressure gradients (in optically thick conditions), neutrino advection, and velocity corrections are taken into account, but transport is performed only in the radial direction. Transport is computed by means of multigroup flux-limited diffusion with a flux limiter that has been tuned to reproduce Boltzmann transport results to within a few percent [94]. All O(v/c) observer corrections have been included. The transport solver is fully implicit and solves for four neutrino flavors simultaneously (i.e., $\nu_{\rm e}$'s, $\bar{\nu}_{\rm e}$'s, ν_μ's and ν_τ's (collectively $\nu_{\rm x}$'s), and $\bar{\nu}_\mu$'s and $\bar{\nu}_\tau$'s (collectively $\bar{\nu}_{\rm x}$'s)), allowing for neutrino-neutrino scattering and pair-exchange, and different ν and $\bar{\nu}$ opacities. The PPM technology has been directly applied to both the spatial and energy advection of neutrinos in both the radial and lateral directions.

Below we derive the fully general relativistic 1D MGFLD equations which we solve along each radial ray.. These can be solved in a number of levels of GR approximations, depending on how the metric functions, a, and b (see below) are computed. For a purely Newtonian calculations both of these metric parameters are set to unity. Up to now we have implemented a approximation suggested by [98] in which a is approximated by a suitable gravitational potential and b is left equal to unity.

0.9.2 Neutrino Boltzmann Equation

The general form of the radiation Boltzmann equation for spherically symmetric spacetimes with momentum components measured in an orthonormal frame comoving with the matter has been given by [96, 37, 110, 108, 10]. We follow [96] and [108] and begin with the Boltzmann equation written in an arbitrary coordinate basis

$$\frac{\partial f_i(x,p)}{\partial x^\alpha}\frac{dx^\alpha}{d\lambda} + \frac{\partial f_i(x,p)}{\partial p^\beta}\frac{dp^\beta}{d\lambda} = \left(\frac{d}{d\lambda}f_i(x,p)\right)_{\rm sources}. \tag{33}$$

Here p^β is the particle four-momentum defined by

$$p^\beta = \frac{dx^\beta}{d\lambda} \tag{34}$$

where λ is the affine path length. The invariant distribution function, f_i, is defined such that

$$dN_i = \frac{1}{h^3} f_i(x,p)(p\cdot n) dV dP \tag{35}$$

where dN_i is the number of neutrinos of flavor i in the coordinate invariant volume elements dV and dP, defined by

$$dV = \sqrt{-g}\epsilon_{\lambda\alpha\beta\gamma} n^\lambda d_1x^\alpha d_2x^\beta d_3x^\gamma \tag{36}$$

where n is a unit timelike vector (orthogonal to the volume element by construction), g is the determinant of the metric, $\epsilon_{\lambda\alpha\beta\gamma}$ is the Levi-Civita unit antisymmetric symbol ($\epsilon_{0123} = +1$), and

$$dP = \sqrt{-g}\epsilon_{ijk}\frac{d_1p^i d_2p^j d_3p^k}{-p_0}. \tag{37}$$

We assume that the neutrinos follow null geodesics in between localized interactions, so that

$$\frac{dp^\alpha}{d\lambda} = -\left\{ \begin{array}{cc} \alpha \\ \beta & \gamma \end{array} \right\} p^\beta p^\gamma, \tag{38}$$

where $\left\{ \begin{array}{cc} \alpha \\ \beta & \gamma \end{array} \right\}$ is the Christoffel symbol of the second kind. (This Christoffel symbol is sometimes denoted by $\Gamma^\alpha_{\beta\gamma}$, but we will use the latter symbol to denote the Ricci rotation coefficients, which are the connection coefficients for the case in which the vector basis is chosen to be an orthonormal basis rather than a coordinate basis.) Using equations (34) and (38), equation (33) can be rewritten as

$$p^\alpha\left[\frac{\partial f_i}{\partial x^\alpha} - \left\{ \begin{array}{cc} \beta \\ \alpha & \gamma \end{array} \right\} p^\gamma \frac{\partial f_i}{\partial p^\beta}\right] = \left(\frac{d}{d\lambda}f_i(x,p)\right)_{\text{sources}}. \tag{39}$$

The localized interactions are described by the term on the right-hand side of equation (39).

We now specialize to an orthogonal spherical polar coordinate basis ($\mathbf{e_t}, \mathbf{e}_\xi, \mathbf{e}_\theta, \mathbf{e}_\phi$) with a "synchronous" metric given by

$$\begin{aligned} ds^2 &= g_{\alpha\beta}dx^\alpha dx^\beta \\ &= a^2\left(\frac{x^0}{c}, x^1\right)(dx^0)^2 - b^2\left(\frac{x^0}{c}, x^1\right)(dx^1)^2 - R^2\left(\frac{x^0}{c}, x^1\right)\left[(dx^2)^2 + \sin^2\theta(dx^3)^2\right] \\ &= a(t,\xi)^2c^2dt^2 - b(t,\xi)^2d\xi^2 - R(t,\xi)^2\left(d\theta^2 + \sin^2\theta d\phi^2\right), \end{aligned} \tag{40}$$

where R is the areal radius (2-sphere $\text{area} = 4\pi R^2$), $a = a(t,\xi)$ is the lapse function, and $b = b(t,\xi)$ will be chosen below such that the radial coordinate ξ is a Lagrangian coordinate equal to the enclosed rest mass.

To tie the radial coordinate ξ to the comoving frame, i.e., make it a Lagrangian variable, we use the condition that in the comoving frame

$$u^\nu \equiv \frac{dx^\nu}{d\tau} \equiv c\frac{dx^\nu}{ds} = [u^0, 0, 0, 0]. \tag{41}$$

where $d\tau = adt$. It follows from this and the form of the metric given by equation (40) that

$$u^0 = \frac{c}{a} \tag{42}$$

Now let $\rho = \rho(t, \xi)$ be the proper rest mass density. We will assume that rest mass is not created or destroyed so that ρ satisfies the local conservation condition

$$0 = (\rho u^\nu)_{;\nu} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial x^\nu} (\sqrt{g} \rho u^\nu) \tag{43}$$

Use of condition (43) along with equations (40) and (42) gives

$$\frac{\partial}{\partial t} \left(bR^2 \rho \right) = 0 \tag{44}$$

or

$$\left(bR^2 \rho \right) = f_i(\xi) \tag{45}$$

where $f_i(\xi)$ is an arbitrary function of ξ. Observing that a spherical shell of thickness $d\xi$ has proper volume $dV = 4\pi R^2 \, b \, d\xi$, the mass dm enclosed by the shell is given by $dm = 4\pi R^2 \rho \, b \, d\xi$. We choose $f_i(\xi) = \frac{1}{4\pi}$ so that equation (45) gives

$$b = \frac{1}{4\pi R^2 \rho}. \tag{46}$$

and therefore

$$d\xi = dm. \tag{47}$$

This fixes the Lagrangian radial coordinate as the enclosed rest mass. Having chosen the Lagrangian radial coordinate thus, we will hereafter denote it by m rather than by ξ.

With unit vector n oriented so that $n^0 = 1/a$, $n^i = 0$, then dV as given by equation (36) can be written

$$dV = bR^2 dm \, d\Omega = bR^2 \frac{dm}{dR} dR \, d\Omega = \frac{1}{\Gamma} 4\pi R^2 dR \, d\Omega, \tag{48}$$

where $\Gamma \equiv \frac{1}{b} \frac{\partial R}{\partial m}$ is given below by equation (126).

The source functions are most easily evaluated if the components of vectors and tensors (in particular, the components of the neutrino momenta) are measured in a comoving, orthonormal frame. We choose the orthonormal basis vectors $(\mathbf{e}_{\hat{t}}, \mathbf{e}_{\hat{m}}, \mathbf{e}_{\hat{\theta}}, \mathbf{e}_{\hat{\phi}})$ to be parallel to the coordinate basis, but rescaled such that $(\mathbf{e}_{\hat{\mathrm{a}}})^\nu (\mathbf{e}_{\hat{\mathrm{b}}})_\nu = \eta_{\hat{\mathrm{a}}\hat{\mathrm{b}}}$, where $\eta_{\hat{\mathrm{a}}\hat{\mathrm{b}}} = \mathrm{diag}(1, -1, -1, -1)$ is the Minkowski metric, while $(\mathbf{e}_\alpha)^\nu (\mathbf{e}_\beta)_\nu = g_{\alpha\beta}$. Here we use the convention, when needed to avoid ambiguity, that labels of basis vectors and indices in the orthonormal frame are denoted by lower case Latin letters adorned with a hat while labels of basis vectors and indices in the coordinate basis are denoted by lower case Greek letters. The transformation coefficients, $\epsilon^\alpha{}_{\hat{\mathrm{a}}}$ and $\epsilon^{\hat{\mathrm{a}}}{}_\alpha$, relating the two bases are defined by

$$\mathbf{e}_{\hat{a}} = \epsilon^\alpha{}_{\hat{\mathrm{a}}} \mathbf{e}_\alpha \qquad \mathbf{e}_\alpha = \epsilon^{\hat{\mathrm{a}}}{}_\alpha \mathbf{e}_{\hat{a}}. \tag{49}$$

In our case we have $\epsilon^{\hat{0}}{}_0 = (\epsilon^0{}_{\hat{0}})^{-1} = c\,a(t,m)$, $\epsilon^{\hat{1}}{}_1 = (\epsilon^1{}_{\hat{1}})^{-1} = b(t,m)$, $\epsilon^{\hat{2}}{}_2 = (\epsilon^2{}_{\hat{2}})^{-1} = R$, and $\epsilon^{\hat{3}}{}_3 = (\epsilon^3{}_{\hat{3}})^{-1} = R\sin\theta$, with all others being zero. In the orthonormal basis, the Boltzmann equation becomes

$$p^{\hat{a}}\left[\frac{\partial f_i}{\partial x^{\hat{a}}} - \Gamma^{\hat{b}}_{\hat{a}\hat{c}}p^{\hat{c}}\frac{\partial f_i}{\partial p^{\hat{b}}}\right] = \left(\frac{d}{d\lambda}f_i(x,p)\right)_{\text{sources}}, \tag{50}$$

where $\Gamma^{\hat{b}}_{\hat{a}\hat{c}}$ are the Ricci rotation coefficients defined by

$$\Gamma^{\hat{b}}_{\hat{a}\hat{c}} \equiv \epsilon^{\alpha}{}_{\hat{a}}\epsilon^{\hat{b}}{}_{\beta}\epsilon^{\beta}{}_{\hat{c};\alpha} = \epsilon^{\alpha}{}_{\hat{a}}\epsilon^{\hat{b}}{}_{\beta}\left[\partial_{\alpha}\epsilon^{\beta}{}_{\hat{c}} + \left\{\begin{array}{cc}\beta\\ \alpha & \gamma\end{array}\right\}\epsilon^{\gamma}{}_{\hat{c}}\right]. \tag{51}$$

The metric (40) gives rise to the following nonzero Christoffel symbols:

$$\left\{\begin{array}{cc}0\\ 0 & 0\end{array}\right\} = \frac{a_{,t}}{a}, \quad \left\{\begin{array}{cc}0\\ 0 & 1\end{array}\right\} = \left\{\begin{array}{cc}0\\ 1 & 0\end{array}\right\} = \frac{a_{,m}}{a} \tag{52}$$

$$\left\{\begin{array}{cc}0\\ 1 & 1\end{array}\right\} = \frac{bb_{,t}}{c^2a^2}, \quad \left\{\begin{array}{cc}0\\ 2 & 2\end{array}\right\} = \frac{RR_{,t}}{c^2a^2}, \quad \left\{\begin{array}{cc}0\\ 3 & 3\end{array}\right\} = \frac{RR_{,t}\sin^2\theta}{c^2a^2} \tag{53}$$

$$\left\{\begin{array}{cc}1\\ 0 & 0\end{array}\right\} = \frac{c^2aa_{,m}}{b^2}, \quad \left\{\begin{array}{cc}1\\ 1 & 1\end{array}\right\} = \frac{b_{,m}}{b}, \quad \left\{\begin{array}{cc}1\\ 2 & 2\end{array}\right\} = -\frac{RR_{,m}}{b^2}, \quad \left\{\begin{array}{cc}1\\ 3 & 3\end{array}\right\} = -\frac{RR_{,m}\sin^2\theta}{b^2} \tag{54}$$

$$\left\{\begin{array}{cc}1\\ 1 & 0\end{array}\right\} = \left\{\begin{array}{cc}1\\ 0 & 1\end{array}\right\} = \frac{b_{,t}}{b}, \quad \left\{\begin{array}{cc}2\\ 2 & 0\end{array}\right\} = \left\{\begin{array}{cc}2\\ 0 & 2\end{array}\right\} = \left\{\begin{array}{cc}3\\ 3 & 0\end{array}\right\} = \left\{\begin{array}{cc}3\\ 0 & 3\end{array}\right\} = \frac{R_{,t}}{R} \tag{55}$$

$$\left\{\begin{array}{cc}2\\ 1 & 2\end{array}\right\} = \left\{\begin{array}{cc}2\\ 2 & 1\end{array}\right\} = \left\{\begin{array}{cc}3\\ 1 & 3\end{array}\right\} = \left\{\begin{array}{cc}3\\ 3 & 1\end{array}\right\} = \frac{R_{,m}}{R} \tag{56}$$

$$\left\{\begin{array}{cc}2\\ 3 & 3\end{array}\right\} = -\sin\theta\cos\theta, \quad \left\{\begin{array}{cc}3\\ 2 & 3\end{array}\right\} = \left\{\begin{array}{cc}3\\ 3 & 2\end{array}\right\} = \cot\theta \tag{57}$$

where $x_{,t} \equiv \left(\frac{\partial x}{\partial t}\right)_m$ and $x_{,m} \equiv \left(\frac{\partial x}{\partial m}\right)_t$ for any variable x. Using the expressions for the $\epsilon^{\hat{a}}{}_{\alpha}$'s given below equation (49) and equations (52) - (57) for the Christoffel symbols, we find that the nonzero Ricci rotation coefficients (i.e., connection coefficients in our orthormal basis) are

$$\Gamma^{\hat{0}}_{\hat{0}\hat{1}} = \Gamma^{\hat{1}}_{\hat{0}\hat{0}} = \frac{a_{,m}}{ab} \tag{58}$$

$$\Gamma^{\hat{0}}_{\hat{1}\hat{1}} = \Gamma^{\hat{1}}_{\hat{1}\hat{0}} = \frac{1}{ac}\frac{b_{,t}}{b} \tag{59}$$

$$\Gamma^{\hat{0}}_{\hat{2}\hat{2}} = \Gamma^{\hat{0}}_{\hat{3}\hat{3}} = \Gamma^{\hat{2}}_{\hat{2}\hat{0}} = \Gamma^{\hat{3}}_{\hat{3}\hat{0}} = \frac{1}{ac}\frac{R_{,t}}{R} \tag{60}$$

$$\Gamma^{\hat{2}}_{\hat{2}\hat{1}} = \Gamma^{\hat{3}}_{\hat{3}\hat{1}} = -\Gamma^{\hat{1}}_{\hat{2}\hat{2}} = -\Gamma^{\hat{1}}_{\hat{3}\hat{3}} = \frac{1}{b}\frac{R_{,m}}{R} \tag{61}$$

$$\Gamma^{\hat{3}}_{\hat{3}\hat{2}} = -\Gamma^{\hat{2}}_{\hat{3}\hat{3}} = \frac{\cot\theta}{R} \tag{62}$$

The components $p^{\hat{\mathrm{a}}}$ of the neutrino four-momentum are taken with respect to the comoving orthonormal basis $(\mathbf{e}_{\hat{t}}, \mathbf{e}_{\hat{m}}, \mathbf{e}_{\hat{\theta}}, \mathbf{e}_{\hat{\phi}})$ and are given by

$$(p^{\hat{0}}, p^{\hat{1}}, p^{\hat{2}}, p^{\hat{3}}) \equiv (p^{\hat{0}}, \bar{p}) = \frac{1}{c}\left[\epsilon, \epsilon\mu, \epsilon\sqrt{1-\mu^2}\cos\phi_p, \epsilon\sqrt{1-\mu^2}\sin\phi_p\right] \tag{63}$$

where the quantities ϵ, μ, and ϕ_p are measured by an observer comoving with the fluid, $\epsilon = cp^{\hat{0}} = cp^{\alpha}u_{\alpha}$ being the neutrino energy, $\mu = \cos\theta_p$ the cosine of the angle of $\bar{p}$ relative to the radial direction $\mathbf{e}_{\hat{m}}$, and ϕ_p the azimuthal angle measured from $\mathbf{e}_{\hat{\theta}}$. We take the $p^{\hat{1}}$, $p^{\hat{2}}$, and $p^{\hat{3}}$ components as independent components, with $p^{\hat{0}}$ being a function of these through the mass shell relation $p_{\hat{\mathrm{a}}}p^{\hat{\mathrm{a}}} = 0$. The invariant distribution function is therefore a function of the $p^{\hat{i}}$ but not explicitly of $p^{\hat{0}}$. The sum over $\hat{\mathrm{b}}$ in equation (50), consequently, runs from from $\hat{\mathrm{b}} = 1$ to $\hat{\mathrm{b}} = 3$. It follows from equations (62) that

$$\frac{\partial}{\partial p^{\hat{1}}} = c\mu\frac{\partial}{\partial\epsilon} + \frac{c(1-\mu^2)}{\epsilon}\frac{\partial}{\partial\mu}, \tag{64}$$

$$\frac{\partial}{\partial p^{\hat{2}}} = c(1-\mu^2)^{1/2}\cos(\phi_p)\left(\frac{\partial}{\partial\epsilon} - \frac{\mu}{\epsilon}\frac{\partial}{\partial\mu}\right), \tag{65}$$

$$\frac{\partial}{\partial p^{\hat{3}}} = c(1-\mu^2)^{1/2}\sin(\phi_p)\left(\frac{\partial}{\partial\epsilon} - \frac{\mu}{\epsilon}\frac{\partial}{\partial\mu}\right). \tag{66}$$

Using equations (58) – (66) in equation (50), we obtain the general relativistic neutrino Boltzmann equation for the metric (40)

$$\begin{aligned}
&\frac{\epsilon}{c}\left\{\frac{1}{c}\left(\frac{\partial f_i}{\partial\tau}\right)_{m,\epsilon,\mu} + \frac{\mu}{b}\left(\frac{\partial f_i}{\partial m}\right)_{t,\epsilon,\mu}\right.\\
&-\frac{\epsilon}{c}\left[\frac{a_{,m}c}{ab}\mu + \frac{b_{,\tau}}{b}\mu^2 + \frac{R_{,\tau}}{R}(1-\mu^2)\right]\left(\frac{\partial f_i}{\partial\epsilon}\right)_{t,m,\mu}\\
&\left.+\left[\mu\left(\frac{1}{c}\frac{R_{,\tau}}{R} - \frac{b_{,\tau}}{cb}\right) - \frac{a_{,m}}{ab} + \frac{1}{b}\frac{R_{,m}}{R}\right](1-\mu^2)\left(\frac{\partial f_i}{\partial\mu}\right)_{t,m,\epsilon}\right\}\\
&= \left(\frac{d}{d\lambda}f_i(x,p)\right)_{\text{sources}} \equiv \text{RHS}.
\end{aligned} \tag{67}$$

where we have set $d\tau = adt$, and used the notation $x_{\tau} \equiv \frac{\partial x}{\partial\tau}$. [Here care us required with this notation. For example, $x_{\tau m}$ is not a mixed partial in the strict sense, since derivatives are taken with respect to constant t, not τ. In particular, $x_{\tau m} \neq x_{m\tau}$.]

0.9.3 Neutrino Boltzmann Equation in Terms of Independent Variables $(t, m, a\epsilon, \mu)$

The moment equations derived from equation (67) can be numerically solved more accurately if we switch from independent variables (t, m, ϵ, μ) to independent variables $(t, m, a\epsilon, \mu)$. With this variable transformation, and with $f_i(t, m, \mu, \epsilon) \rightarrow f_i(t, m, \mu, a\epsilon)$, the neutrino Boltzmann equation takes the form

$$\begin{aligned}
&\frac{\epsilon}{c}\left\{\frac{1}{c}\left(\frac{\partial f_i}{\partial \tau}\right)_{m,a\epsilon,\mu} + \frac{\mu}{b}\left(\frac{\partial f_i}{\partial m}\right)_{t,a\epsilon,\mu}\right. \\
&-\frac{1}{c}\left[-\frac{a_{,\tau}}{a} + \frac{b_{,\tau}}{b}\mu^2 + \frac{R_{,\tau}}{R}(1-\mu^2)\right] a\epsilon\left(\frac{\partial f_i}{\partial (a\epsilon)}\right)_{t,m,\mu} \\
&\left.+\left[\mu\left(\frac{1}{c}\frac{R_{,\tau}}{R} - \frac{b_{,\tau}}{cb}\right) - \frac{a_{,m}}{ab} + \frac{1}{b}\frac{R_{,m}}{R}\right](1-\mu^2)\left(\frac{\partial f_i}{\partial \mu}\right)_{t,m,a\epsilon}\right\} \\
&= \left(\frac{d}{d\lambda} f_i(x,p)\right)_{\text{sources}} \equiv \text{RHS}. \qquad (68)
\end{aligned}$$

The benefit of this transformation is that the radial derivative of a in the factor multiplying the energy derivative of f_i in equation (67) has been transformed into a time derivative. Thus all terms in the coefficient multiplying the energy derivative of f_i in the transformed equation (68) contain terms involving only time derivatives, and this coefficient therefore vanishes for a static spacetime. The consequence of this is that for a static spacetime there will be no flow of neutrinos through the neutrino energy grid (e.g., due to gravitational redshifting) if the grid energies, ϵ_k (see Appendix B for our grid indexing conventions), are chosen to be functions of t and m such that $a(t, m)\epsilon(t, m)_k = \text{constant}$. If the lapse function $a(t, m)$ is normalized to unity very far from the star, then this choice of grid energies can be written

$$a(t, m)\epsilon(t, m)_k(t, m) = \epsilon_{k\,\infty} \qquad (69)$$

where the $\epsilon_{k\,\infty}$'s are the grid energy very far from the star. For the nonstatic spacetimes encountered during the shock reheating epoch of a core collapse supernova, the term $\frac{a_{,m}c}{ab}$ multiplying $\left(\frac{\partial f_i}{\partial(\epsilon)}\right)_{t,m,\mu}$ in the original equation (67) is typically two orders of magnitude larger than the term $\frac{a_{,t}}{a}$ that replaces it in equation (68). During the shock reheating epoch, therefore, the flow of neutrinos through the energy grid in equation (68) will be at a much smaller rate than in the original equation (67) and consequently far easier to handle numerically. Futhermore, we will show below that on differencing equation (68) the contribution of the energy derivative term can be conveniently operator split from the other terms and treated separately.

0.9.4 Moment Equations

To derive equations satisfied by angular moments of the distribution function, we introduce the following comoving variables. Define the comoving n^{th} angular moment

$\psi_i^{(n)}$ of the invariant distribution function by

$$\psi_i^{(n)}(t,m,a\epsilon) \equiv \frac{1}{4\pi}\int \mu^n d\Omega f_i(t,m,\mu,a\epsilon) = \frac{1}{2}\int_{-1}^{1} \mu^n d\mu f_i(t,m,\mu,a\epsilon), \quad (70)$$

and define the corresponding comoving $n^{\rm th}$ angular moment $\mathrm{RHS}^{(n)}$ of the source terms RHS by

$$\mathrm{RHS}^{(n)} \equiv \frac{1}{4\pi}\int \mu^n d\Omega\, \mathrm{RHS} = \frac{1}{2}\int_{-1}^{1} \mu^n d\mu\, \mathrm{RHS}, \quad (71)$$

Now, operating on the transport equation (68) by $\frac{1}{2}\int_{-1}^{1} d\mu$ and $\frac{1}{2}\int_{-1}^{1} \mu d\mu$ and using equations (70) and (71), we obtain, respectively, the following equations for the evolution of the zeroth and first comoving angular moments

$$\begin{aligned}
&\frac{\epsilon}{c}\left\{\frac{1}{c}\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)_{m,a\epsilon} + \frac{1}{b}\left(\frac{\partial\psi_i^{(1)}}{\partial m}\right)_{t,a\epsilon}\right.\\
&+\frac{1}{c}a\epsilon\left[\frac{a_{,\tau}}{a}\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m} - \frac{b_{,\tau}}{b}\left(\frac{\partial\psi_i^{(2)}}{\partial(a\epsilon)}\right)_{t,m} - \frac{R_{,\tau}}{R}\left(\frac{\partial}{\partial(a\epsilon)}\right)_{t,m}\left(\psi_i^{(0)} - \psi_i^{(2)}\right)\right]\\
&\left.-\left(\frac{1}{c}\frac{R_{,\tau}}{r} - \frac{b_{,\tau}}{cb}\right)\left(\psi_i^{(0)} - 3\psi_i^{(2)}\right) + \left(-\frac{a_{,m}}{ab} + \frac{1}{b}\frac{R_{,m}}{R}\right)2\psi_i^{(1)}\right\}\\
&= \mathrm{RHS}^{(0)}
\end{aligned} \quad (72)$$

and

$$\begin{aligned}
&\frac{\epsilon}{c}\left\{\frac{1}{c}\left(\frac{\partial\psi_i^{(1)}}{\partial\tau}\right)_{m,a\epsilon} + \frac{1}{b}\left(\frac{\partial\psi_i^{(2)}}{\partial m}\right)_{t,a\epsilon}\right.\\
&+\frac{1}{c}a\epsilon\left[\frac{a_{,\tau}}{a}\left(\frac{\partial\psi_i^{(1)}}{\partial(a\epsilon)}\right)_{t,m} - \frac{b_{,\tau}}{b}\left(\frac{\partial\psi_i^{(3)}}{\partial(a\epsilon)}\right)_{t,m} - \frac{R_{,\tau}}{R}\left(\frac{\partial}{\partial(a\epsilon)}\right)_{t,m}\left(\psi_i^{(1)} - \psi_i^{(3)}\right)\right]\\
&\left.-\left(\frac{1}{c}\frac{R_{,\tau}}{R} - \frac{b_{,\tau}}{cb}\right)\left(2\psi_i^{(1)} - 4\psi_i^{(3)}\right) - \left(-\frac{a_{,m}}{ab} + \frac{1}{b}\frac{R_{,m}}{R}\right)(\psi_i^{(0)} - 3\psi_i^{(2)})\right\}\\
&= \mathrm{RHS}^{(1)}
\end{aligned} \quad (73)$$

In order to derive the Einstein equations below, we will need the radiation stress-energy tensor, $\mathcal{T}$. To construct the covariant radiation stress-energy tensor, we begin with its definition (e.g., see Mihalas & Mihalas 1984)

$$\mathcal{T}^{\alpha\beta} = \frac{c^2}{h^3}\int \sum_i f_i p^\alpha p^\beta \frac{d^3p}{cp^0}. \quad (74)$$

where the sum over i is a sum over all neutrino types. In the local comoving orthonormal frame, the components of the stress-energy tensor become

$$\mathcal{T}^{\hat{a}\hat{b}} = \frac{c^2}{(hc)^3} \int \sum_i f_i p^{\hat{a}} p^{\hat{b}} \epsilon d\epsilon d\Omega. \tag{75}$$

Using equation (62) for $p^{\hat{a}}$ and $p^{\hat{b}}$ in equation (74), and noting that the local neutrino energy density, E_ν, flux, F_ν, and pressure, P_ν, are given by

$$E_\nu = \frac{1}{(hc)^3} \int \sum_i f_i \epsilon^3 d\epsilon d\Omega = \frac{4\pi}{(hc)^3} \int \sum_i \psi_i^{(0)} \epsilon^3 d\epsilon, \tag{76}$$

$$F_\nu = \frac{c}{(hc)^3} \int \sum_i f_i \epsilon^3 d\epsilon \mu d\Omega = \frac{4\pi c}{(hc)^3} \int \sum_i \psi_i^{(1)} \epsilon^3 d\epsilon, \tag{77}$$

and

$$P_\nu = \frac{1}{(hc)^3} \int \sum_i f_i \epsilon^3 d\epsilon \mu^2 d\Omega = \frac{4\pi}{(hc)^3} \int \sum_i \psi_i^{(2)} \epsilon^3 d\epsilon, \tag{78}$$

the components of $\mathcal{T}$ given by equation (75), transformed back to the coordinate basis, can be written

$$\mathcal{T}_\alpha{}^\beta = \begin{pmatrix} E_\nu & \frac{a}{b} F_\nu & 0 & 0 \\ -\frac{b}{ca^2} F_\nu & -P_\nu & 0 & 0 \\ 0 & 0 & -\frac{1}{2}(E_\nu - P_\nu) & 0 \\ 0 & 0 & 0 & -\frac{1}{2}(E_\nu - P_\nu) \end{pmatrix} \tag{79}$$

We will use these below in deriving the Einstein equations.

0.9.5 Flux-Limiting

The two radiation moment equations (72) and (73) contain the three unknown radiation moments $\psi_i^{(0)}$, $\psi_i^{(1)}$, and $\psi_i^{(2)}$, and an additional equation must therefore be supplied for closure. A practicable and robust method for accomplishing this which preserves causality in optically thin regions is flux-limiting. Our approach here in deriving the flux-limiting diffusion equations is noncovariant (but applicable to our chosen Lagrangian frame) and follows in outline the method of Levermore (1984).

Define $\eta_i^{(i)}$, $i \geq 1$ by

$$\psi_i^{(i)} = \eta_i^{(i)} \psi_i^{(0)}, \tag{80}$$

Substituting equations (80) in equations (72) and (73), gives

$$\frac{1}{ac}\left[\left(\frac{\partial\psi_i^{(0)}}{\partial t}\right)_{m,a\epsilon}+\frac{a_{,t}}{a}a\epsilon\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}\right]+\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}\eta_i^{(1)}+\frac{1}{b}\left(\frac{\partial\eta_i^{(1)}}{\partial m}\right)_{t,a\epsilon}\psi_i^{(0)}$$
$$-\frac{\epsilon}{c}\left[\frac{b_{,t}}{b}\left(\frac{\partial\eta_i^{(2)}}{\partial(a\epsilon)}\right)_{t,m}\psi_i^{(0)}+\frac{b_{,t}}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}\eta_i^{(2)}\right.$$
$$\left.+\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(0)}-\eta_i^{(2)}\right)}{\partial(a\epsilon)}\right)_{t,m}\psi_i^{(0)}+\frac{R_{,t}}{R}\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}\left(\eta_i^{(0)}-\eta_i^{(2)}\right)\right]$$
$$-\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\left(\eta_i^{(0)}-3\eta_i^{(2)}\right)\psi_i^{(0)}+\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)2\eta_i^{(1)}\psi_i^{(0)}$$
$$=\frac{c}{\epsilon}\,\mathrm{RHS}^{(0)},\tag{81}$$

and

$$\frac{1}{ac}\left[\left(\frac{\partial\eta_i^{(1)}}{\partial t}\right)_{m,a\epsilon}\psi_i^{(0)}+\left(\frac{\partial\psi_i^{(0)}}{\partial t}\right)_{m,a\epsilon}\eta_i^{(1)}+\frac{a_{,t}}{a}a\epsilon\left(\frac{\partial\eta_i^{(1)}}{\partial(a\epsilon)}\right)_{t,m}\psi_i^{(0)}\right.$$
$$\left.+\frac{a_{,t}}{a}a\epsilon\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}\eta_i^{(1)}\right]+\frac{1}{b}\left(\frac{\partial\eta_i^{(2)}}{\partial m}\right)_{t,a\epsilon}\psi_i^{(0)}+\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}\eta_i^{(2)}$$
$$-\frac{\epsilon}{c}\left[\frac{b_{,t}}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}\eta_i^{(3)}+\frac{b_{,t}}{b}\left(\frac{\partial\eta_i^{(3)}}{\partial(a\epsilon)}\right)_{t,m}\psi_i^{(0)}\right.$$
$$\left.+\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(1)}-\eta_i^{(3)}\right)}{\partial(a\epsilon)}\right)_{t,m}\psi_i^{(0)}+\frac{R_{,t}}{R}\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}\left(\eta_i^{(1)}-\eta_i^{(3)}\right)\right]$$
$$-\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\left(2\eta_i^{(1)}-4\eta_i^{(3)}\right)\psi_i^{(0)}-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)(1-3\eta_i^{(2)})\psi_i^{(0)}$$
$$=\frac{c}{\epsilon}\,\mathrm{RHS}^{(1)}.\tag{82}$$

Solving equation (81) for $\frac{1}{ac}\left(\frac{\partial\psi_i^{(0)}}{\partial t}\right)_{r,a\epsilon}$, substituting the result into equation (82) and factoring out $\psi_i^{(0)}$ gives

$$\left\{ -\frac{\eta_i^{(1)}}{b}\left(\frac{\partial \eta_i^{(1)}}{\partial m}\right)_{t,a\epsilon} + \eta_i^{(1)}\frac{\epsilon}{c}\frac{b_{,t}}{b}\left(\frac{\partial \eta_i^{(2)}}{\partial (a\epsilon)}\right)_{t,m} + \eta_i^{(1)}\frac{\epsilon}{c}\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(0)}-\eta_i^{(2)}\right)}{\partial(a\epsilon)}\right)_{t,m} \right.$$
$$+\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\eta_i^{(1)}\left(\eta_i^{(0)}-3\eta_i^{(2)}\right)-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)2(\eta_i^{(1)})^2+\frac{1}{ac}\left(\frac{\partial \eta_i^{(1)}}{\partial t}\right)_{m,a\epsilon}$$
$$+\frac{1}{ac}\frac{a_{,t}}{a}a\epsilon\left(\frac{\partial \eta_i^{(1)}}{\partial (a\epsilon)}\right)_{t,m}+\frac{1}{b}\left(\frac{\partial \eta_i^{(2)}}{\partial m}\right)_{t,a\epsilon}-\frac{\epsilon}{c}\frac{b_{,t}}{b}\left(\frac{\partial \eta_i^{(3)}}{\partial (a\epsilon)}\right)_{t,m}-\frac{\epsilon}{c}\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(1)}-\eta_i^{(3)}\right)}{\partial(a\epsilon)}\right)_{t,m}$$
$$\left.-\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\left(2\eta_i^{(1)}-4\eta_i^{(3)}\right)-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)\left(1-3\eta_i^{(2)}\right)\right\}\psi_i^{(0)}$$
$$+\frac{1}{b}\left(\frac{\partial \psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}\eta_i^{(2)}-\frac{1}{b}\left(\frac{\partial \psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}(\eta_i^{(1)})^2+\eta_i^{(1)}\frac{c}{\epsilon}\,\mathrm{RHS}^{(0)}$$
$$+\left\{\frac{b_{,t}}{b}\left(\eta_i^{(1)}\eta_i^{(2)}-\eta_i^{(3)}\right)+\frac{R_{,t}}{R}\left[\eta_i^{(1)}\left(\eta_i^{(0)}-\eta_i^{(2)}-1\right)+\eta_i^{(3)}\right]\right\}\frac{\epsilon}{c}\left(\frac{\partial \psi_i^{(0)}}{\partial (a\epsilon)}\right)_{t,m}$$
$$=\frac{c}{\epsilon}\,\mathrm{RHS}^{(1)}. \tag{83}$$

Consider the factor multiplying $\psi_i^{(0)}$ in equation (83) in the diffusion and free streaming limits. In the diffusion limit,

$$\eta_i^{(1)}\to 0;\qquad \eta_i^{(2)}\to\frac{1}{3};\qquad \eta_i^{(3)}\to 0, \tag{84}$$

so that

$$-\frac{\eta_i^{(1)}}{b}\left(\frac{\partial \eta_i^{(1)}}{\partial m}\right)_{t,a\epsilon}+\eta_i^{(1)}\frac{\epsilon}{c}\frac{b_{,t}}{b}\left(\frac{\partial \eta_i^{(2)}}{\partial (a\epsilon)}\right)_{t,m}+\eta_i^{(1)}\frac{\epsilon}{c}\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(0)}-\eta_i^{(2)}\right)}{\partial(a\epsilon)}\right)_{t,m}$$
$$+\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\eta_i^{(1)}\left(\eta_i^{(0)}-3\eta_i^{(2)}\right)-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)2(\eta_i^{(1)})^2+\frac{1}{ac}\left(\frac{\partial \eta_i^{(1)}}{\partial t}\right)_{m,a\epsilon}$$
$$+\frac{1}{ac}\frac{a_{,t}}{a}a\epsilon\left(\frac{\partial \eta_i^{(1)}}{\partial (a\epsilon)}\right)_{t,m}+\frac{1}{b}\left(\frac{\partial \eta_i^{(2)}}{\partial m}\right)_{t,a\epsilon}-\frac{\epsilon}{c}\frac{b_{,t}}{b}\left(\frac{\partial \eta_i^{(3)}}{\partial (a\epsilon)}\right)_{t,m}-\frac{\epsilon}{c}\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(1)}-\eta_i^{(3)}\right)}{\partial(a\epsilon)}\right)_{t,m}$$
$$-\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\left(2\eta_i^{(1)}-4\eta_i^{(3)}\right)-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)\left(1-3\eta_i^{(2)}\right)$$
$$\to 0, \tag{85}$$

where we have set all the derivatives of $\eta_i^{(i)}$ to zero. In the free streaming limit,

$$\eta_i^{(i)}\to 1, \tag{86}$$

so that

$$
\begin{aligned}
&-\frac{\eta_i^{(1)}}{b}\left(\frac{\partial\eta_i^{(1)}}{\partial m}\right)_{t,a\epsilon}+\eta_i^{(1)}\frac{\epsilon}{c}\frac{b_{,t}}{b}\left(\frac{\partial\eta_i^{(2)}}{\partial(a\epsilon)}\right)_{t,m}+\eta_i^{(1)}\frac{\epsilon}{c}\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(0)}-\eta_i^{(2)}\right)}{\partial(a\epsilon)}\right)_{t,m}\\
&+\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\eta_i^{(1)}\left(\eta_i^{(0)}-3\eta_i^{(2)}\right)-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)2(\eta_i^{(1)})^2+\frac{1}{ac}\left(\frac{\partial\eta_i^{(1)}}{\partial t}\right)_{m,a\epsilon}\\
&+\frac{1}{ac}\frac{a_{,t}}{a}a\epsilon\left(\frac{\partial\eta_i^{(1)}}{\partial(a\epsilon)}\right)_{t,m}+\frac{1}{b}\left(\frac{\partial\eta_i^{(2)}}{\partial m}\right)_{t,a\epsilon}-\frac{\epsilon}{c}\frac{b_{,t}}{b}\left(\frac{\partial\eta_i^{(3)}}{\partial(a\epsilon)}\right)_{t,m}-\frac{\epsilon}{c}\frac{R_{,t}}{R}\left(\frac{\partial\left(\eta_i^{(1)}-\eta_i^{(3)}\right)}{\partial(a\epsilon)}\right)_{t,m}\\
&-\left(\frac{1}{ac}\frac{R_{,t}}{R}-\frac{b_{,t}}{acb}\right)\left(2\eta_i^{(1)}-4\eta_i^{(3)}\right)-\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)\left(1-3\eta_i^{(2)}\right)\\
&\qquad\qquad\rightarrow 0,
\end{aligned}
\tag{87}
$$

where again we have set all the derivatives of $\eta_i^{(i)}$ to zero. The factor multiplying $\psi_i^{(0)}$ in equation (83) is thus zero both in the diffusion limit and in the free streaming limit. This is also true of the factor multiplying $\frac{\epsilon}{c}\left(\frac{\partial\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}$. We make the approximation that these two factors are zero everywhere. Equation (83) then becomes

$$
\frac{c}{\epsilon}\,\mathrm{RHS}^{(0)}\eta_i^{(1)}+\frac{1}{b}\left[\eta_i^{(2)}-(\eta_i^{(1)})^2\right]\left(\frac{\partial\psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}=\frac{c}{\epsilon}\,\mathrm{RHS}^{(1)}. \tag{88}
$$

Factoring out $\eta_i^{(1)}$ from $\mathrm{RHS}^{(1)}$, i.e.,

$$
\mathrm{RHS}^{(1)}=\mathrm{rhs}^{(1)}\eta_i^{(1)} \tag{89}
$$

and using equation (89) in equation (88) gives

$$
\frac{c}{\epsilon}\,\mathrm{RHS}^{(0)}\eta_i^{(1)}+\left(\eta_i^{(2)}-(\eta_i^{(1)})^2\right)\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}=\frac{c}{\epsilon}\,\mathrm{rhs}^{(1)}\eta_i^{(1)}. \tag{90}
$$

Solving equation (90) for $\eta_i^{(1)}$ and multiplying by $\psi_i^{(0)}$, gives the diffusion-like equation

$$
\psi_i^{(1)}=\psi_i^{(0)}\eta_i^{(1)}=-\frac{\left(\eta_i^{(2)}-(\eta_i^{(1)})^2\right)\psi_i^{(0)}}{\frac{c}{\epsilon}\left(\mathrm{RHS}^{(0)}-\mathrm{rhs}^{(1)}\right)}\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}. \tag{91}
$$

We now define $\lambda_i^{(t)}$ as

$$
\lambda_i^{(t)}=\frac{\psi_i^{(0)}}{\frac{c}{\epsilon}\left(\mathrm{RHS}^{(0)}-\mathrm{rhs}^{(1)}\right)}, \tag{92}
$$

and regard $[\eta_i^{(2)} - (\eta_i^{(1)})^2]$ as a free parameter, which we write as

$$\mathcal{F}_i = 3[\eta_i^{(2)} - (\eta_i^{(1)})^2]. \tag{93}$$

Using equations (92) and (93) in equation (91), we get

$$\psi_i^{(1)} = \eta_i^{(1)}\psi_i^{(0)} = -\frac{\lambda_i^{(t)}}{3}\mathcal{F}_i\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}}{\partial m}\right)_{t,a\epsilon}, \tag{94}$$

where equation (94) has the form of a diffusion equation and $\mathcal{F}_i$ is referred to as the "flux-limiter". $\mathcal{F}_i$ is unity in the diffusion limit, and should tend to zero in such a way that $\psi_i^{(1)} = \psi_i^{(0)}$ in the limit of radial free streaming. Equations (72) and (94) for each energy zone, with a prescription for $\mathcal{F}_i$, constitute the GR MGFLD equations. The prescription for the flux-limiter $\mathcal{F}_i$ will be given below. At the edge of the computational grid, the interior metric should be joined to a metric appropriate for neutrinos radiating radially away in a vacuum, i.e., a Vadya metric. However, we include enough of the mantle of the models in our calculations such that u of the outer boundary is small and the exterior metric is essentially flat. We therefore set a and Γ equal to 1 at the outer boundary with negligible error.

0.9.6 Matter–Neutrino Energy–Momentum Exchange

Our GR MGFLD equations, consisting of equations (72) and (94) for each energy zone, must be supplemented by equations describing the energy-momentum and lepton exchange with the matter. The former is described by the GR hydrodynamics equation, given by

$$T^{\beta}_{\alpha\ ;\beta} = G_\alpha \tag{95}$$

where the left-hand side is the divergence of the energy-momentum tensor of the matter, and G^α is the four-force density, i.e., the neutrino $\rightarrow$ matter energy-momentum transfer rate per unit volume. An expression for G^α is given by the integral of the product of the invariant momentum volume element, the neutrino four-momentum, and the invariant source terms of the transport equation summed over all neutrino flavors

$$G^\alpha = -\frac{c}{h^3}\int\frac{\sqrt{-g}d^3p}{p_0}\ p^\alpha\ \sum_i \mathrm{RHS}_i. \tag{96}$$

That equation (96) is correct can be ascertained by transforming the operator $-\frac{c}{h^3}\int\frac{\sqrt{-g}d^3p}{p_0}\ p^\alpha\ \sum_i$ to the orthonrmal basis and operating on equation (68). The right-hand side equation (68) then gives the right-hand side of equation (96) in the orthonormal basis, or $G^{\hat{a}}$. The left-hand side is, apart from transport terms, given by

$$
\begin{aligned}
&-\frac{\epsilon}{c}\frac{c}{(hc)^3}\int\frac{\epsilon^2 d\epsilon d\Omega_p}{\epsilon/c}\frac{\epsilon}{c}\left(1,\mu,(1-\mu^2)^{1/2}\cos\phi_p,(1-\mu^2)^{1/2}\sin\phi_p\right)\left(\frac{\partial f_i}{\partial\tau}\right)_{m,a\epsilon,\mu}\\
&=-\frac{1}{(hc)^3}\int\epsilon^2 d\epsilon d\Omega_p\left(\epsilon,\mu\epsilon,\epsilon(1-\mu^2)^{1/2}\cos\phi_p,\epsilon(1-\mu^2)^{1/2}\sin\phi_p\right)\left(\frac{\partial f_i}{\partial\tau}\right)_{m,a\epsilon,\mu}\\
&=-\frac{2\pi}{(hc)^3}\int\epsilon^3 d\epsilon d\mu\left[\left.\frac{\partial}{\partial\tau}\right|_{m,a\epsilon,\mu}(f_i,\mu f_i,0,0)\right]\\
&=\left[-\left(\frac{\partial E_\nu}{\partial\tau}\right)_{m,a\epsilon,\mu},-c\left(\frac{\partial P_\nu}{\partial\tau}\right)_{m,a\epsilon,\mu},0,0\right] \qquad (97)
\end{aligned}
$$

which is the negative of the local rate of change of the neutrino energy–momentum flux per unit volume, and therefore local energy–momentum transfer rate per unit volume for the matter as asserted. The four-force density in the orthonormal frame is given by

$$
\begin{aligned}
\{G^{\hat{\alpha}}\}&=-\frac{c}{(hc)^3}\int\frac{\epsilon^2 d\epsilon d\Omega_p}{\epsilon/c}\,\frac{\epsilon}{c}\left(1,\mu,(1-\mu^2)^{1/2}\cos\phi_p,(1-\mu^2)^{1/2}\sin\phi_p\right)\,\sum_i \mathrm{RHS}_i\\
&=-\frac{4\pi c}{(hc)^3}\int\epsilon^2 d\epsilon\,\left(\sum_i \mathrm{RHS}_i^{(0)},\sum_i \mathrm{RHS}_i^{(1)},0,0\right) \qquad (98)
\end{aligned}
$$

which in the coordinate basis becomes

$$
G^\alpha=-\frac{4\pi c}{(hc)^3}\int\epsilon^2 d\epsilon\,\left(\frac{1}{a}\sum_i \mathrm{RHS}_i^{(0)},\frac{1}{b}\sum_i \mathrm{RHS}_i^{(1)},0,0\right) \qquad (99)
$$

Turning to the left-hand side of equation (95), the matter energy-momentum tensor is given by the perfect fluid expression

$$
T^{\alpha\beta}=(\rho c^2+E_{\mathrm{m}}+P_{\mathrm{m}})\frac{u^\alpha u^\beta}{c^2}-P_{\mathrm{m}}g^{\alpha\beta} \qquad (100)
$$

where ρ, E_{m}, and P_{m} are, respectively, the rest mass density, internal energy density, and pressure of the matter (all evaluated in the rest frame of the matter).

The energy equation is obtained by projecting equation (95) along the fluid four-velocity:

$$
u^\alpha T^{\beta}_{\alpha\ ;\beta}=u^\alpha G_\alpha. \qquad (101)
$$

Using equation (99) for G^α and equations (41) and (42) for u^α in the comoving frame, the right-hand side of equation (101) becomes

$$
u^\alpha G_\alpha=\frac{4\pi c^2}{(hc)^3}\int\epsilon^2 d\epsilon\,\sum_i \mathrm{RHS}_i^{(0)} \qquad (102)
$$

Using equation (100) for $T^{\alpha\beta}$ in the left-hand side of (101), and assuming conservation of rest mass [$(\rho u^{\beta})_{;\beta} = 0$], we get

$$u^{\alpha} T^{\beta}_{\alpha\ ;\beta} = E_{\mathrm{m},\beta} u^{\beta} + (E_{\mathrm{m}} + p_{\mathrm{m}}) u^{\beta}_{;\beta}. \tag{103}$$

Denoting by e_{m} the internal energy per unit rest mass ($e_{\mathrm{m}} = E_{\mathrm{m}}/\rho$) and using $u^{\beta}_{;\beta} = -u^{\beta}\rho_{,\beta}/\rho$, equation (103) becomes

$$u^{\alpha} T^{\beta}_{\alpha\ ;\beta} = \rho e_{\mathrm{m},\beta} u^{\beta} - p_{\mathrm{m}} u^{\beta} \frac{\rho_{,\beta}}{\rho} = \frac{1}{a}\frac{\partial e_{\mathrm{m}}}{\partial t} - \frac{c p_{\mathrm{m}}}{a\rho}\frac{\partial \rho}{\partial t}. \tag{104}$$

Equations (102) and (104) in (101) gives us our energy equation

$$\left(\frac{\partial e_{\mathrm{m}}}{\partial \tau}\right)_m = -p_{\mathrm{m}} \left(\frac{\partial\left(\frac{1}{\rho}\right)}{\partial \tau}\right)_m - \frac{4\pi c^2}{(hc)^3}\int \epsilon^2 d\epsilon \sum_i \mathrm{RHS}_i^{(0)} \tag{105}$$

which says that changes in the matter internal energy arise from work due to local compression or expansion and from energy exchange with neutrinos.

The other nontrivial equation in (95) is the radial equation

$$T^{1\beta}_{\ \ ;\beta} = G^1 \tag{106}$$

The left-hand side of (106) is straightforwardly evaluated to give

$$T^{1\beta}_{\ \ ;\beta} = T^{1\beta}_{\ \ ,\beta} + \left\{ \begin{array}{cc} 1 \\ \mu & \beta \end{array} \right\} T^{\mu\beta} + \left\{ \begin{array}{cc} \beta \\ \mu & \beta \end{array} \right\} T^{1\mu} = \frac{1}{b^2}\left(P_{\mathrm{m},m} + \frac{a_{,m}}{a}\rho c^2 w\right) \tag{107}$$

where the "relativistic enthalpy", w, is defined by

$$w \equiv 1 + \frac{e_{\mathrm{m}}}{c^2} + \frac{P_{\mathrm{m}}}{\rho c^2}. \tag{108}$$

Using equations (107) and (99) in (106) gives us

$$P_{\mathrm{m},m} + \frac{a_{,m}}{a}\rho c^2 w = -\frac{4\pi b}{(hc)^3}\int \epsilon^2 d\epsilon \sum_i \mathrm{RHS}_i^{(1)} \tag{109}$$

which relates the spatial gradient of P_{m} to the gradient of the lapse function a and the rate of neutrino-matter momentum exchange. Further elaboration of this equation requires the Einstein equations, which relate the metric functions to the stress-energy of the matter and neutrinos. Equation (109), solved for $\frac{a_{,m}}{a}$, will be regarded as an equation to be solved for a:

$$\frac{a_{,m}}{a} = -\frac{1}{\rho c^2 w} p_{\mathrm{m},m} - \frac{1}{\rho c^2 w}\frac{4\pi c b}{(hc)^3}\int \epsilon^2 d\epsilon \sum_{\nu} \mathrm{RHS}^{(1)}. \tag{110}$$

0.9.7 Einstein Equations

The Einstein field equations are given by

$$-\frac{8\pi G}{c^4}(T_\nu{}^\mu + \mathcal{T}_\nu{}^\mu) = \mathcal{R}_\nu{}^\mu - \frac{1}{2}g_\nu{}^\mu \mathcal{R}_\lambda{}^\lambda \tag{111}$$

where $\mathcal{R}_\nu{}^\mu$ is the Ricci tensor, and $T_\nu{}^\mu$ and $\mathcal{T}_\nu{}^\mu$ are, respectively, the matter and radiation stress-energy tensors given by equations (100) and (79). The nonzero components of the Ricci tensor are

$$\mathcal{R}_{00} = \frac{b_{,tt}}{b} + 2\frac{R_{,tt}}{R} - \frac{aa_{,mm}c^2}{b^2} + \frac{aa_{,m}b_{,m}c^2}{b^3} - \frac{a_{,t}b_{,t}}{ab} - 2\frac{a_{,t}R_{,t}}{aR} - 2\frac{aa_{,m}c^2R_{,m}}{b^2R} \tag{112}$$

$$\mathcal{R}_{11} = \frac{a_{,mm}}{a} + 2\frac{R_{,mm}}{R} - \frac{bb_{,tt}}{a^2c^2} + \frac{a_{,t}b_{,t}b}{a^3c^2} - 2\frac{b_{,t}bR_{,t}}{a^2c^2R} - \frac{b_{,m}a_{,m}}{ba} - 2\frac{b_{,m}R_{,m}}{bR} \tag{113}$$

$$\mathcal{R}_{01} = 2\frac{R_{,tm}}{R} - 2\frac{a_{,m}R_{,t}}{aR} - 2\frac{b_{,t}R_{,m}}{bR}, \tag{114}$$

$$\mathcal{R}_{22} = \mathcal{R}_{33} = -1 - \frac{R_{,tt}R}{a^2c^2} + \frac{a_{,t}R_{,t}R}{a^3c^2} + \frac{R_{,mm}R}{b^2} - \frac{b_{,m}R_{,m}R}{b^3} - \frac{b_{,t}R_{,t}R}{a^2bc^2} - \frac{R_{,t}^2}{a^2c^2} + \frac{a_{,m}R_{,m}R}{ab^2} + \frac{R_{,m}^2}{b^2} \tag{115}$$

and the Ricci scalar is given by

$$\mathcal{R} = \mathcal{R}_\kappa{}^\kappa = g^{\mu\kappa}\mathcal{R}_{\mu\kappa}$$
$$= 2\frac{b_{,tt}}{a^2bc^2} + 4\frac{R_{,tt}}{a^2c^2R} - 2\frac{a_{,mm}}{ab^2} + 2\frac{a_{,m}b_{,m}}{ab^3} - 2\frac{a_{,t}b_{,t}}{a^3bc^2} - 4\frac{a_{,t}R_{,t}}{a^3Rc^2} - 4\frac{a_{,m}R_{,m}}{ab^2R}$$
$$-4\frac{R_{,mm}}{b^2R} + 4\frac{b_{,t}R_{,t}}{a^2bc^2R} + 4\frac{b_{,m}R_{,m}}{b^3R} + 2\frac{R_{,t}^2}{a^2c^2R^2} - 2\frac{R_{,m}^2}{b^2R^2} + \frac{2}{R^2}. \tag{116}$$

Using equations (112) - (116), (100), and (79) in equation (111), the Einstein equations are given by

$$\binom{1}{0} \qquad \frac{R_{,\tau m}}{R} - \frac{a_{,m}R_{,\tau}}{aR} - \frac{b_{,\tau}R_{,m}}{bR} = \frac{4\pi G}{c^4}(bF_\nu) \tag{117}$$

$$\binom{0}{0} \qquad \frac{c^2}{2}\left[R + \frac{RR_{,\tau}^2}{c^2} - \frac{RR_{,m}^2}{b^2}\right]_{,m} = R^2R_{,\tau}\frac{4\pi G}{c^4}(bF_\nu) + R^2R_{,m}\frac{4\pi G}{c^2}(\rho c^2 + E_{\rm m} + P_{\rm m}) \tag{118}$$

$$\binom{1}{1} \qquad \frac{c^2}{2}\left[R + \frac{RR_{,\tau}^2}{c^2} - \frac{RR_{,m}^2}{b^2}\right]_{,\tau} = -\frac{4\pi G}{c^2}R^2R_{,m}\frac{1}{b}F_\nu - \frac{4\pi G}{c^2}R^2R_{,\tau}(P_{\rm m} + P_\nu). \tag{119}$$

where equation (117) has been used to get the forms of equations (118) and (119).

Defining

$$\Gamma = \frac{R_{,m}}{b} = 4\pi R^2 \rho R_{,m}, \tag{120}$$

$$u = \frac{R_{,t}}{a} = R_{,\tau}, \tag{121}$$

$$GM_{\rm g} \equiv GM_{\rm g}(t,m) \equiv \frac{c^2}{2}\left[R + R\frac{u^2}{c^2} - R\Gamma^2\right] \tag{122}$$

Equations (118) and 119) can be written

$$M_{\rm g\,,m} = \left[(\rho + \frac{E_{\rm m}}{c^2} + \frac{E_\nu}{c^2}) + \frac{1}{c^4}\frac{uF_\nu}{\Gamma}\right] 4\pi R^2 R_{,m} \tag{123}$$

$$M_{\rm g\,,\tau} = -\frac{4\pi R^2}{c^2}\left[u(P_{\rm m} + P_\nu) + \Gamma F_\nu\right]. \tag{124}$$

Equation (123) integrates to

$$\begin{aligned} M_{\rm g} = M_{\rm g}(t,m) &= \int_0^m \left[(\rho + \frac{E_{\rm m}}{c^2} + \frac{E_\nu}{c^2}) + \frac{1}{c^4}\frac{uF_\nu}{\Gamma}\right] 4\pi R^2 R_{,m} dm \\ &= \int_0^m \left[(\rho + \frac{E_{\rm m}}{c^2} + \frac{E_\nu}{c^2}) + \frac{1}{c^4}\frac{uF_\nu}{\Gamma}\right] \frac{\Gamma}{\rho} dm. \end{aligned} \tag{125}$$

and equation (122) can be solved for Γ to give

$$\Gamma = \sqrt{1 + \frac{u^2}{c^2} - \frac{2M_{\rm g}G}{Rc^2}} \tag{126}$$

The $\binom{1}{1}$ Einstein equations can be used to derive an equation of motion. Expanding the derivative with respect to τ on the left-hand side, using the $\binom{1}{0}$ equation for $R_{\tau m}$, and using the "momentum" equation (110) for $a_{,m}$, we get

$$u_{,\tau} = -G\frac{M_{\rm g}}{R^2} - 4\pi R^2 \frac{\Gamma}{w} p_{{\rm m},m} - \frac{4\pi G}{c^2} R(P_{\rm m} + P_\nu) - \frac{\Gamma}{\rho w}\frac{4\pi c}{(hc)^3}\int \epsilon^2 d\epsilon \sum_\nu {\rm RHS}(1) \tag{127}$$

Finally, an equation for the change in the density, ρ, can be obtained by solving the $\binom{1}{0}$ Einstein equation for b_τ/b, and expressing b_τ/b by use of equation (46). This gives

$$\frac{(\rho R^2)_{,\tau}}{\rho R^2} = -\frac{u_{,m}}{R_{,m}} + \frac{R}{R_{,m}}\frac{4\pi G}{c^4}(bF_\nu) \tag{128}$$

0.9.8 Matter–Neutrino Lepton Number Exchange

The local lepton number density of the matter is given by

$$n_{\mathrm{e}^-} - n_{\mathrm{e}^+} = \frac{\rho}{m_{\mathrm{B}}} Y_{\mathrm{e}} \tag{129}$$

where n_{p}, n_{B}, n_{e^-}, and n_{e^+} are the proper proton, baryon, electron, and positron number densities, respectively, Y_{e} is the proton fraction, and m_{B} is the mean baryon mass. An equation describing the evolution of the matter lepton number is obtained by equating its four-divergence to its sources in the ν_{e} and $\bar{\nu}_{\mathrm{e}}$ fields:

$$\left(\frac{\rho}{m_{\mathrm{B}}} Y_{\mathrm{e}} u^{\alpha}\right)_{;\alpha} = -\frac{c}{h^3} \int \frac{\sqrt{-g} d^3p}{p_0} \left(\mathrm{RHS}_{\nu_{\mathrm{e}}} - \mathrm{RHS}_{\bar{\nu}_{\mathrm{e}}}\right) \tag{130}$$

where $\mathrm{RHS}_{\nu_{\mathrm{e}}}$ and $\mathrm{RHS}_{\bar{\nu}_{\mathrm{e}}}$ are the source terms for the ν_{e} and $\bar{\nu}_{\mathrm{e}}$ transport equations, respectively. Assuming rest mass conservation, the left-hand side of equation (130) becomes

$$\left(\frac{\rho}{m_{\mathrm{B}}} Y_{\mathrm{e}} u^{\alpha}\right)_{;\alpha} = \frac{\rho}{m_{\mathrm{B}}} \frac{1}{a} \frac{\partial Y_{\mathrm{e}}}{\partial t}. \tag{131}$$

Evaluating the RHS in the orthonormal frame and noting that the only terms which do not vanish upon integrating over solid angle and neutrino number are emission and absorption, we get

$$\begin{aligned} -\frac{c}{h^3} \int \frac{\sqrt{-g} d^3p}{p_0} \left(\mathrm{RHS}_{\nu_{\mathrm{e}}} - \mathrm{RHS}_{\bar{\nu}_{\mathrm{e}}}\right) &= -\frac{c^2}{(hc)^3} \int \epsilon d\epsilon d\Omega \ \left(\mathrm{RHS}_{\nu_{\mathrm{e}}} - \mathrm{RHS}_{\bar{\nu}_{\mathrm{e}}}\right) \\ &= -\frac{4\pi c}{(hc)^3} \int \epsilon^2 d\epsilon \left[\left(j_{\nu_{\mathrm{e}}}(\epsilon) \left(1 - \psi^{(0)}_{\nu_{\mathrm{e}}}(\epsilon)\right) - \frac{\psi^{(0)}_{\nu_{\mathrm{e}}}(\epsilon)}{\lambda_{\nu_{\mathrm{e}}}(\epsilon)}\right)\right. \\ &\quad \left. - \left(j_{\bar{\nu}_{\mathrm{e}}}(\epsilon) \left(1 - \psi^{(0)}_{\bar{\nu}_{\mathrm{e}}}(\epsilon)\right) - \frac{\psi^{(0)}_{\bar{\nu}_{\mathrm{e}}}(\epsilon)}{\lambda_{\bar{\nu}_{\mathrm{e}}}(\epsilon)}\right)\right]. \end{aligned} \tag{132}$$

where $j_{\nu_{\mathrm{e}}}(\epsilon)$ and $j_{\bar{\nu}_{\mathrm{e}}}(\epsilon)$ are ν_{e} and $\bar{\nu}_{\mathrm{e}}$ emisisivities (in inverse mean free paths), and $\lambda_{\nu_{\mathrm{e}}}(\epsilon)$ and $\lambda_{\bar{\nu}_{\mathrm{e}}}(\epsilon)$ the corresponding absorption mean free paths. Equations (131) and (132) in (130) gives an equation for the evolution of the matter lepton fraction:

$$\begin{aligned} \frac{\partial Y_{\mathrm{e}}}{\partial t} = -\frac{4\pi a c}{(hc)^3} \frac{m_{\mathrm{B}}}{\rho} \int \epsilon^2 d\epsilon &\left[\left(j_{\nu_{\mathrm{e}}}(\epsilon) \left(1 - \psi^{(0)}_{\nu_{\mathrm{e}}}(\epsilon)\right) - \frac{\psi^{(0)}_{\nu_{\mathrm{e}}}(\epsilon)}{\lambda_{\nu_{\mathrm{e}}}(\epsilon)}\right)\right. \\ &\left. - \left(j_{\bar{\nu}_{\mathrm{e}}}(\epsilon) \left(1 - \psi^{(0)}_{\bar{\nu}_{\mathrm{e}}}(\epsilon)\right) - \frac{\psi^{(0)}_{\bar{\nu}_{\mathrm{e}}}(\epsilon)}{\lambda_{\bar{\nu}_{\mathrm{e}}}(\epsilon)}\right)\right] \end{aligned} \tag{133}$$

0.9.9 Summary of Equations

Here we summarize the equations derived in the preceding sections, suppressing the (t, m) dependence of variables. The MGFLD equations comprise the continuity-like equation for the zeroth angular moment of the neutrino occupation number [equation (72]

$$\begin{aligned}
&\frac{\epsilon}{c}\left\{\frac{1}{c}\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial\tau}\right)_{m,a\epsilon}+\frac{1}{b}\left(\frac{\partial\psi_i^{(1)}(a\epsilon)}{\partial m}\right)_{t,a\epsilon}\right.\\
&+\frac{1}{c}a\epsilon\left[\frac{a_{,\tau}}{a}\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial(a\epsilon)}\right)_{t,m}-\frac{b_{,\tau}}{b}\left(\frac{\partial\psi_i^{(2)}(a\epsilon)}{\partial(a\epsilon)}\right)_{t,m}-\frac{R_{,\tau}}{R}\left(\frac{\partial}{\partial(a\epsilon)}\right)_{t,m}\left(\psi_i^{(0)}(a\epsilon)-\psi_i^{(2)}(a\epsilon)\right)\right]\\
&\left.-\left(\frac{1}{c}\frac{R_{,\tau}}{r}-\frac{b_{,\tau}}{cb}\right)\left(\psi_i^{(0)}(a\epsilon)-3\psi_i^{(2)}(a\epsilon)\right)+\left(-\frac{a_{,m}}{ab}+\frac{1}{b}\frac{R_{,m}}{R}\right)2\psi_i^{(1)}(a\epsilon)\right\}\\
&=\mathrm{RHS}^{(0)}
\end{aligned}\tag{134}$$

the diffusion-like equation for first angular moment of the neutrino occupation number [equation (94)]

$$\psi_i^{(1)}(a\epsilon)=-\frac{\lambda_i^{(t)}(a\epsilon)}{3}\mathcal{F}_i(a\epsilon)\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial m}\right)_{t,a\epsilon}=-\frac{\lambda_i^{(t)}(a\epsilon)}{3}\Gamma\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial R}\right)_{t,a\epsilon},\tag{135}$$

the Eddington closure relation for the second moment, discussed below, needed to close equation (134)

$$\psi_i^{(2)}(a\epsilon)=\mathcal{E}_i(\psi_i^{(0)}(a\epsilon),\psi_i^{(1)}(a\epsilon)),\tag{136}$$

the equations of matter motion including the neutrino–matter momentum exchange [equations (128), (127), (121), (108), (125)] and neutrino–matter energy exchange [equation (105)]

$$\frac{(\rho R^2)_{,\tau}}{\rho R^2}=-\frac{u_{,m}}{R_{,m}}+\frac{R}{R_{,m}}\frac{4\pi G}{c^4}(bF_\nu)\tag{137}$$

$$u_{,\tau}=-G\frac{M_\mathrm{g}}{R^2}-4\pi R^2\frac{\Gamma}{w}p_{\mathrm{m},m}-\frac{4\pi G}{c^2}R(P_\mathrm{m}+P_\nu)-\frac{\Gamma}{\rho w}\frac{4\pi c}{(hc)^3}\int\epsilon^2 d\epsilon\sum_i\mathrm{RHS}_i^{(1)}\tag{138}$$

$$R_{,\tau}=u\tag{139}$$

$$w=1+\frac{e_\mathrm{m}}{c^2}+\frac{p_\mathrm{m}}{\rho c^2},\tag{140}$$

$$M_{\rm g} = \int_0^m \left[(\rho + \frac{E_{\rm m}}{c^2} + \frac{E_\nu}{c^2}) + \frac{1}{c^4}\frac{uF_\nu}{\Gamma} \right] \frac{\Gamma}{\rho} dm, \tag{141}$$

$$\left(\frac{\partial e_{\rm m}}{\partial \tau}\right)_m = -p_{\rm m} \left(\frac{\partial\left(\frac{1}{\rho}\right)}{\partial \tau}\right)_m - \frac{4\pi c^2}{(hc)^3} \int \epsilon^2 d\epsilon \sum_i {\rm RHS}_i^{(0)} \tag{142}$$

the lepton conservation equation [equation (133)]

$$\frac{\partial Y_{\rm e}}{\partial \tau} = -\frac{4\pi c}{(hc)^3}\frac{m_{\rm B}}{\rho} \int \epsilon^2 d\epsilon \left[\left(j_{\nu_{\rm e}}(a\epsilon)\left(1 - \psi^{(0)}_{\nu_{\rm e}}(a\epsilon)\right) - \frac{\psi^{(0)}_{\nu_{\rm e}}(a\epsilon)}{\lambda_{\nu_{\rm e}}(a\epsilon)} \right) - \left(j_{\bar\nu_{\rm e}}(a\epsilon)\left(1 - \psi^{(0)}_{\bar\nu_{\rm e}}(a\epsilon)\right) - \frac{\psi^{(0)}_{\bar\nu_{\rm e}}(a\epsilon)}{\lambda_{\bar\nu_{\rm e}}(a\epsilon)} \right) \right] \tag{143}$$

and the equations for the space-time [equations (120), (126), and (110)]

$$\Gamma = \frac{R_{,m}}{b} = 4\pi R^2 \rho R_{,m} = \sqrt{1 + \frac{u^2}{c^2} - \frac{2M_{\rm g}G}{Rc^2}} \tag{144}$$

$$\frac{a_{,m}}{a} = -\frac{1}{\rho c^2 w} p_{{\rm m},m} - \frac{1}{\rho c^2 w}\frac{4\pi c b}{(hc)^3} \int \epsilon^2 d\epsilon \sum_\nu {\rm RHS}^{(1)}, \tag{145}$$

0.9.10 The equations presently used in CHIMERA

The equations derived above are a complete set of fully general relativistic radiation-hydrodynamic equations for spherical symmetry. At the present time CHIMERA is not fully general relativistic because it is built to simulate non-spherically symmetric problems. CHIMERA therefore uses the transport subset of equations (134)-(145), namely, equations (134), (135), 142), and (143), and couples them to multidimensional PPM Newtonian hydrodynamics, described elsewhere in this report.

CHIMERA currently supports two levels of approximations to fully GR radiation-hydrodynamics. The first is a fully Newtonian mode. In this mode the metric parameters, a and b, in equations (134), (135), 142), and (143) are set equal to unity and these equations then reduce to their order v/c special relativistic counterparts. The gravitational is computed by a spectral Poisson solver as described in [117] and is of the form

$$\Phi(R_j, \theta_o, \phi_p) = -G \sum_{\ell=1}^{L} \frac{4\pi}{2\ell+1} \sum_{m=-\ell}^{\ell} Y^{\ell m}(\theta_o, \phi_p) \left(\frac{1}{R_j^{\ell+1}} C_n^{\ell m} + R_j D_n^{\ell m} \right) \tag{146}$$

The second mode is a post-Newtonain approximation in which the spherically symmetric component of the gravitational potential is modified as described by [98] to the form

$$\Phi_{\rm PN\ spherical} = -G \int_R^\infty \frac{dR'}{R^2} \left(\frac{M_G}{4\pi} + R'^3 (p + p_\nu)/c^2 \right) \frac{1}{\Gamma^2} \left(\frac{\rho + (e+p)/c^2}{c^2} \right) \tag{147}$$

where p and e are the material pressure and internal energy per unit volume, p_ν is the neutrino pressure,

$$M_G = 4\pi \int_0^R dR' R'^2 \Gamma \left(\rho + (e + e_\nu)/c^2 + \frac{u f_\nu}{\Gamma c^4} \right) \tag{148}$$

$$\Gamma = \sqrt{1 + \frac{u^2}{c^2} - \frac{2 M_G G}{R c^2}} \tag{149}$$

and e_ν and f_ν are the neutrino energy and flux, respectively. The gravitational potential used in this approximation is then

$$\Phi(R_j, \theta_o, \phi_p) + \Phi_{\rm PN\ spherical} - \Phi_{\rm N\ spherical} \tag{150}$$

where $\Phi_{\rm N\ spherical}$ is the Newtonian spherical gravitational potential computed by

$$\Phi_{\rm N\ spherical} = -G \int_R^\infty \frac{dR'}{R^2} \left(\frac{M_N}{4\pi} \right) \tag{151}$$

with

$$M_N = 4\pi \int_0^R dR' R'^2 \rho \tag{152}$$

The lapse function a in this approximation is computed by

$$a = \sqrt{1 + \frac{2\Phi_{\rm PN\ spherical}}{c^2}} \tag{153}$$

and is used in both the hydro and the transport equations. In the latter it accounts for gravitational time dilation and redshifting. The metric function b is left equal to unity. A comparison of newtonian and GR spherically symmetric core collapse [26] showed that the metric function b does not play an important role in neutrino transport.

0.9.11 Differencing of the MGFLD Equations

Here we will describe our method of differencing the MGFLD equations, which consist of equations (134), (135), (136), (142) (with the matter pressure term operator split off), and (143). The hydrodynamic equations and the equations for the metric functions (to various levels of approximation) operator split and explicitly differenced. The MGFLD equations, on the other hand, are implicitly differenced as described below.

To write the difference equations, we will adopt the following grid indexing convention. Subscripts $j-\frac{1}{2}$ and $j+\frac{1}{2}$ will denote the lower and upper boundaries of the $j^{\rm th}$ radial zone, and j will denote the corresponding zone center. Likewise, $k-\frac{1}{2}$ and $k+\frac{1}{2}$ will denote the lower and upper boundaries of the $k^{\rm th}$ energy zone, and k will denote the corresponding energy zone center. The time at the end of the $n^{\rm th}$ and $(n+1)^{\rm th}$ time steps, and the time midway between the $n^{\rm th}$ and $(n+1)^{\rm th}$ time step will be denoted by superscripts n, $n+1$, and $n+\frac{1}{2}$, respectively. Thus, the quantity $\psi_{j,k}^{(0)\ n+1}$ denotes the comoving zeroth angular moment of the neutrino distribution function at the center of radial zone j, the center of energy zone k, and at the end of the $(n+1)^{th}$ time step.

The Transport Equations

To difference equations (134), (135), (136), (142), and (143) we begin by writing (134) in the following conservative form

$$\frac{1}{ac}\frac{a^3}{bR^2}\left(\frac{\partial(\frac{bR^2}{a^3}\psi_i^{(0)})}{\partial t}\right)_{m,a\epsilon}+\frac{1}{b}\frac{a^2}{R^2}\left(\frac{\partial(\frac{R^2}{a^2}\psi_i^{(1)})}{\partial m}\right)_{t,a\epsilon}$$
$$+\frac{1}{(a\epsilon)^2}\frac{1}{ac}\left[\left(\frac{\dot a}{a}-\frac{\dot R}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m}-\left(\frac{\dot b}{b}-\frac{\dot R}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(2)}}{\partial(a\epsilon)}\right)_{t,m}\right]$$
$$=\mathrm{RHS}_i^{(0)}, \tag{154}$$

or switching variables from t to the proper time τ, where $d\tau=adt$,

$$\frac{1}{c}\frac{a^3}{bR^2}\left(\frac{\partial(\frac{bR^2}{a^3}\psi_i^{(0)})}{\partial \tau}\right)_{m,a\epsilon}+\frac{1}{b}\frac{a^2}{R^2}\left(\frac{\partial(\frac{R^2}{a^2}\psi_i^{(1)})}{\partial m}\right)_{\tau,a\epsilon}$$
$$+\frac{1}{(a\epsilon)^2}\frac{1}{c}\left[\left(\frac{a_\tau}{a}-\frac{R_\tau}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{\tau,m}-\left(\frac{b_\tau}{b}-\frac{R_\tau}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(2)}}{\partial(a\epsilon)}\right)_{\tau,m}\right]$$
$$=\mathrm{RHS}_i^{(0)}, \tag{155}$$

where $X_\tau\equiv\left(\frac{\partial X}{\partial\tau}\right)_m$. To interpret equation (155), operate by $\frac{4\pi}{(\hbar c)^3}\int_0^\infty\epsilon^2 d\epsilon$ at constant τ amd m to obtain

$$\frac{1}{c}\frac{1}{bR^2}\frac{4\pi}{(\hbar c)^3}\int_0^\infty (a\epsilon)^2 d(a\epsilon)\left(\frac{\partial(\frac{bR^2}{a^3}\psi_i^{(0)})}{\partial\tau}\right)_{m,a\epsilon}$$
$$+\frac{1}{b}\frac{1}{aR^2}\frac{4\pi}{(\hbar c)^3}\int_0^\infty (a\epsilon)^2 d(a\epsilon)\left(\frac{\partial(a\frac{R^2}{a^3}\psi_i^{(1)})}{\partial m}\right)_{\tau,a\epsilon}$$
$$+\frac{1}{a^3 c}\frac{4\pi}{(\hbar c)^3}\int_0^\infty d(a\epsilon)\left[\left(\frac{a_\tau}{a}-\frac{R_\tau}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{\tau,m}-\left(\frac{b_\tau}{b}-\frac{r_\tau}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(2)}}{\partial(a\epsilon)}\right)_{\tau,m}\right]$$
$$=\frac{1}{cbR^2}\left(\frac{\partial(bR^2\frac{4\pi}{(\hbar c)^3}\int_0^\infty \epsilon^2 d\epsilon\psi_i^{(0)})}{\partial\tau}\right)_{m,a\epsilon}+\frac{1}{abR^2}\left(\frac{\partial(aR^2\frac{4\pi}{(\hbar c)^3}\int_0^\infty \epsilon^2 d\epsilon\psi_i^{(1)})}{\partial m}\right)_{\tau,a\epsilon}$$
$$=\frac{4\pi}{(\hbar c)^3}\int_0^\infty \epsilon^2 d\epsilon\,\mathrm{RHS}_i^{(0)} \tag{156}$$

where the order of integration and differentiation has been interchanged in the first two terms of equation (156) (permissible since the integration variable $a\epsilon$ is held constant in the differentiations). Note that the energy derivative terms, sometimes referred to collectively as the "Doppler term", integrate to zero because $(a\epsilon)^3\psi_i^{(i)}|_0^\infty = 0$. The significance of equation (156) can be appreciated by considering the four-vector, J^α, which we define as the product of the invariant volume element on the mass shell, $d\mathcal{P}$, the four-momentum p^α divided by h^3, and the invariant distribution function $f(x,p)$, i.e.,

$$J_i^\alpha = \frac{1}{h^3}\int p^\alpha d\mathcal{P} f_i(x,p). \tag{157}$$

J_i^α is the neutrino number-flux four-vector, as can be seen by evaluating it in our local orthonormal frame, for which we obtain

$$J_i^{\hat{\alpha}} = \frac{4\pi}{(hc)^3}\int \epsilon^2 d\epsilon\left(\psi_i^{(0)},\psi_i^{(1)},0,0\right) = \frac{1}{c}\left(cn_{i\,\nu},\mathcal{F}_{i\,\nu},0,0\right) \tag{158}$$

where $n_{i\,\nu}$ and $\mathcal{F}_{i\,\nu}$ are the local number density and number flux of neutrinos, respectively. In our coordinate basis, equation (158) becomes

$$J_i^\alpha = \frac{4\pi}{(hc)^3}\int \epsilon^2 d\epsilon\left(\frac{1}{a}\psi_i^{(0)},\frac{1}{b}\psi_i^{(1)},0,0\right), \tag{159}$$

Using (159), equation (156) can be written simply as

$$J^\nu_{i\,;\,\nu} = \frac{4\pi}{(\hbar c)^3}\int_0^\infty \epsilon^2 d\epsilon\,\mathrm{RHS}_i^{(0)} \tag{160}$$

and is thus seen as the covariant continuity equation for neutrino number.

We will operator split the contributions to $\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)_{m,a\epsilon}$ in the continuity equation (156) by writing

$$\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)_{m,a\epsilon} = \left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(m)}_{m,a\epsilon} + \left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(\epsilon)}_{m,a\epsilon} \tag{161}$$

where the contribution $\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(m)}_{m,a\epsilon}$ arises from the diffusion and source terms of equation (156) with R, a, and b constant in time, and the contribution $\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(\epsilon)}_{m,a\epsilon}$ arises from the time derivatives of R, a, and b with no diffusion or sources. That is, $\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(m)}_{m,a\epsilon}$ and $\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(\epsilon)}_{m,a\epsilon}$ are given, respectively, by

$$\frac{1}{c}\left(\frac{\partial\psi_i^{(0)}}{\partial\tau}\right)^{(m)}_{m,a\epsilon} + \frac{1}{b}\frac{a^2}{R^2}\left(\frac{\partial(\frac{R^2}{a^2}\psi_i^{(1)})}{\partial m}\right)_{t,a\epsilon} = \mathrm{RHS}_i^{(0)} \tag{162}$$

and

$$\frac{1}{c}\frac{a^3}{bR^2}\left(\frac{\partial(\frac{bR^2}{a^3}\psi_i^{(0)})}{\partial\tau}\right)^{(\epsilon)}_{m,a\epsilon}$$
$$+\frac{1}{(a\epsilon)^2}\frac{1}{ac}\left[\left(\frac{a_\tau}{a}-\frac{R_\tau}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(0)}}{\partial(a\epsilon)}\right)_{t,m} - \left(\frac{b_\tau}{b}-\frac{R_\tau}{R}\right)\left(\frac{\partial(a\epsilon)^3\psi_i^{(2)}}{\partial(a\epsilon)}\right)_{t,m}\right] = 0 \tag{163}$$

Equation (162) together with equation (134) are differenced conservatively and fully implicitly as follows:

$$\frac{1}{c}\frac{\psi_{i\,j,k}^{(0)\,n+1}-\psi_{i\,j,k}^{(0)\,n}}{\tau^{n+1}-\tau^n}+$$
$$+\frac{a_j^2}{\mathrm{Vol}_j}\left[\frac{1}{a_{j+\frac{1}{2}}^2}\mathrm{Area}_{j+\frac{1}{2}}\psi_{i\,j+\frac{1}{2},k}^{(1)\,n+1} - \frac{1}{a_j^2}\mathrm{Area}_{j-\frac{1}{2}}\psi_{i\,j-\frac{1}{2},k}^{(1)\,n+1}\right] = \mathrm{RHS}_{i\,j,k}^{(0)\,n+1}. \tag{164}$$

and

$$\psi_{i\,j+\frac{1}{2},k}^{(1)\,n+1} = -\frac{\lambda_{i\,j+\frac{1}{2},k}^{(t)\,n+1}}{3}\mathcal{F}_{i\,j+\frac{1}{2},k}^{n+1}\Gamma_{j+\frac{1}{2}}^{n+1}\frac{\psi_{i\,j+1,k}^{(0)\,n+1}-\psi_{i\,j,k}^{(0)\,n+1}}{R_{j+1}^{n+1}-R_j^{n+1}}, \tag{165}$$

where $\mathrm{Area}_{j+\frac{1}{2}} = 4\pi R_{j+\frac{1}{2}}^2$ is the proper area of the outer boundary of zone j, $R_j^{n+1} = \frac{1}{2}(R_{j-\frac{1}{2}}^{n+1}+R_{j+\frac{1}{2}}^{n+1})$, and with the aid of equation (46) $\mathrm{Vol}_j = 4\pi b_j R_j^2(m_{j+\frac{1}{2}}-m_{j-\frac{1}{2}}) =$

$(m_{j+\frac{1}{2}} - m_{j-\frac{1}{2}})/\rho_j$ is the proper volume enclosed between zones edges $j-\frac{1}{2}$ and $j+\frac{1}{2}$. To see that equation (164) is conservative, rewrite it as

$$\begin{aligned}
&\frac{1}{a_j^3}\left(\psi_{i\,j,k}^{(0)\,n+1} - \psi_{i\,j,k}^{(0)\,n}\right)\mathrm{Vol}_j \\
&= -\frac{1}{a_{j+\frac{1}{2}}^3}\mathrm{Area}_{j+\frac{1}{2}}\psi_{i\,j+\frac{1}{2},k}^{(1)\,n+1}a_{j+\frac{1}{2}}(t^{n+1}-t^n)c + \frac{1}{a_{j-\frac{1}{2}}^3}\mathrm{Area}_{j-\frac{1}{2}}\psi_{i\,j-\frac{1}{2},k}^{(1)\,n+1}a_{j-\frac{1}{2}}(t^{n+1}-t^n)c \\
&\qquad + a_j(t^{n+1}-t^n)\frac{c}{a_j^3}\mathrm{Vol}_j \times \mathrm{RHS}_{i\,j,k}^{(0)\,n+1}. \qquad (166)
\end{aligned}$$

Now according to equation (69) and the discussion there, a neutrino propagating along a geodesic passing through $R_{j-\frac{1}{2}}$, R_j, $R_{j+\frac{1}{2}}$, and ∞ in a static spacetime will have locally measured energies at these points related by

$$\epsilon_{j-\frac{1}{2}}a_{j-\frac{1}{2}} = \epsilon_j a_j = \epsilon_{j+\frac{1}{2}}a_{j+\frac{1}{2}} = \epsilon_\infty \qquad (167)$$

Thus, multiplying equation (166) by $\frac{1}{(hc)^3}(\epsilon_{k\,\infty})^2\Delta\epsilon_{k\,\infty}$ and using equation (167), we get that

$$\begin{aligned}
&\frac{(\epsilon_{j,k})^2\Delta\epsilon_{j,k}}{(hc)^3}\left(\psi_{i\,j,k}^{(0)\,n+1} - \psi_{i\,j,k}^{(0)\,n}\right)\mathrm{Vol}_j \\
&+\mathrm{Area}_{j+\frac{1}{2}}\frac{c(\epsilon_{j+\frac{1}{2},k})^2\Delta\epsilon_{j+\frac{1}{2},k}}{(hc)^3}\psi_{i\,j+\frac{1}{2},k}^{(1)\,n+1}a_{j+\frac{1}{2}}(t^{n+1}-t^n) \\
&-\mathrm{Area}_{j-\frac{1}{2}}\frac{c(\epsilon_{j-\frac{1}{2},k})^2\Delta\epsilon_{j-\frac{1}{2},k}}{(hc)^3}\psi_{i\,j-\frac{1}{2},k}^{(1)\,n+1}a_{j-\frac{1}{2}}(t^{n+1}-t^n) \\
&= ca_j(t^{n+1}-t^n)\frac{(\epsilon_{j,k})^2\Delta\epsilon_{j,k}}{(hc)^3}\mathrm{Vol}_j \times \mathrm{RHS}_{i\,j,k}^{(0)\,n+1}. \qquad (168)
\end{aligned}$$

Denoting the number of neutrinos in the volume between $j-\frac{1}{2}$ and $j+\frac{1}{2}$ and in the k'th energy bin at time t^n by $N_{i\,j,k}^n$, the net number of these neutrinos crossing outwards through $j+\frac{1}{2}$ per unit proper time by $S_{i\,j-\frac{1}{2},k}$, and the source of these neutrinos (number created in the volume between $j-\frac{1}{2}$ and $j+\frac{1}{2}$ per unit proper time) by $\dot{N}_{i\,j,k}$, equation (168) can be written

$$N_{i\,j,k}^{n+1} - N_{i\,j,k}^n = -S_{i\,j+\frac{1}{2},k}\,\Delta\tau_{j+\frac{1}{2}}^{n+\frac{1}{2}} + S_{i\,j-\frac{1}{2},k}\,\Delta\tau_{j-\frac{1}{2}}^{n+\frac{1}{2}} + \dot{N}_{i\,j,k}\,\Delta\tau_j^{n+\frac{1}{2}} \qquad (169)$$

which is manifestly conservative. Here $\Delta\tau_j^{n+\frac{1}{2}} = a_j(t^{n+1}-t^n)$ is the proper time interval at j corresponding to the coordinate time interval $t^{n+1}-t^n$.

The Flux Limiter

In the MGFLD scheme the monochromatic first angular moment, $\psi^{(1)} = \psi^{(1)}(t, r, a\epsilon)$, of the neutrino distribution function is related to the zero angular moment, $\psi^{(0)} = \psi^{(0)}(t, r, a\epsilon)$, by equation (136) vis a flux-liminter $\mathcal{F}_i(a\epsilon)$, which is a parameterized function which interpolates between the optically thick diffusion regime, where

$$\psi_i^{(1)}(a\epsilon) = -\frac{\lambda_i^{(t)}(a\epsilon)}{3}\frac{1}{b}\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial m}\right)_{t,a\epsilon} = -\frac{\lambda_i^{(t)}(a\epsilon)}{3}\Gamma\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial R}\right)_{t,a\epsilon} \tag{170}$$

and the optically thin free streaming regime, where

$$\psi_i^{(1)}(a\epsilon) = \psi_i^{(0)}(a\epsilon). \tag{171}$$

where $\lambda_i^{(t)}(a\epsilon)$ is the total transport mean free path for neutrinos of flavor i.

The flux limiter, $\mathcal{F}_i(a\epsilon)$ is constructed to satisfy equations (170) and (171) and consists of two parts. The first part is a specific implementation of the usual scheme for interpolating between the optically thick diffusion regime and the optically thin free streaming regime, namely

$$\mathcal{F}_{i\ \mathrm{intrp}}(a\epsilon) = \frac{1}{1 + \frac{1}{3}\lambda_i^{(t)}(a\epsilon)\left[\frac{\left|\Gamma\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial R}\right|}{\psi_i^{(0)}(a\epsilon)}\right]}. \tag{172}$$

From equation (172) and the expression

$$\psi_i^{(1)}(a\epsilon) = -\frac{\lambda_i^{(t)}(a\epsilon)}{3}\mathcal{F}_{i\ \mathrm{intrp}}(a\epsilon)\Gamma\left(\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial R}\right)_{t,a\epsilon}, \tag{173}$$

we see that in optically thick diffusion regime as $\lambda_i^{(t)}(a\epsilon) \to 0$, we have

$$\psi_i^{(1)}(a\epsilon) \to -\frac{\lambda_i^{(t)}(a\epsilon)}{3}\Gamma\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial R} \tag{174}$$

while in the optically thin free-streaming regime as $\lambda_i^{(t)}(a\epsilon) \to \infty$, we have

$$\psi_i^{(1)}(a\epsilon) = \psi_i^{(0)}(a\epsilon). \tag{175}$$

As it stands, this scheme suffers from the generic problem of too rapid a transition to the free streaming limit (i.e., the angular distribution becomes too forwardly peaked) when matter goes from optically thick to optically thin abruptly, as when the“density cliff” forms in the post bounce core of a supernova progenitor. To avoid this problem, a second piece of the flux limiter is constructed. It basically prevents the neutrino

angular distribution from becoming more forwardly peaked than the geometrical limit. This piece is given by the following parameterized function

$$\mathcal{F}_{i\ \mathrm{geom}}(a\epsilon) = \begin{cases} \frac{\frac{1}{2}(1+\mu_{i\ 0}(a\epsilon))\psi_i^{(0)}(a\epsilon)}{\frac{1}{3}\lambda_i^{(t)}(a\epsilon)\left[\frac{\left|\Gamma\frac{\partial\psi_i^{(0)}(a\epsilon)}{\partial R}\right|}{\psi_i^{(0)}(a\epsilon)}\right]} & \text{if } R > R_{i\ \nu}(a\epsilon) \\ 1 & \text{if } R \leq R_{i\ \nu}(a\epsilon) \end{cases} \tag{176}$$

where

$$\mu_{i\ 0}(a\epsilon) = \frac{\mu_i(a\epsilon) + \beta}{1 - \mu_i(a\epsilon)\beta}, \tag{177}$$

$$\mu_i(a\epsilon) = \sqrt{1 - \left(\frac{R_{i\ \nu}(a\epsilon)}{R}\right)^2 \mathcal{G}_i}, \tag{178}$$

and

$$\mathcal{G}_i = \sqrt{\frac{1 - 2GMg/rc^2}{1 - 2GM_g/R_{i\ \nu}(\epsilon)c^2}}, \tag{179}$$

where M_g is the gravitational mass, $R_{i\ \nu}(\epsilon)$ is the radius of the neutrinosphere for neutrinos of flavor i, energy ϵ, R is the circumferential radius of the point in question

$$\beta = \frac{v}{c}. \tag{180}$$

The quantity $\mu_i(a\epsilon)$ is the cosine of the angle from the limb of the neutrinosphere of neutrinos of flavor i, energy ϵ, to the point at R, corrected for gravitational bending, and $\mu_{i\ 0}$ is that same angle as seen in the fluid frame. Note that $\mathcal{F}_{i\ \mathrm{geom}}(a\epsilon)$ also interpolates between the diffusion and free streaming regimes.

The net flux limiter, $\mathcal{F}_i$, is given by

$$\mathcal{F}_i(a\epsilon) = \min\left[\mathcal{F}_{i\ \mathrm{intrp}}(a\epsilon), \mathcal{F}_{i\ \mathrm{geom}}(a\epsilon)\right] \tag{181}$$

The Eddington Factor

The Eddington factor given by equation (136) is a closure relation relating the second angular moment of the neutrino occupation number to the first and zeroth angular moment. It must have the property that in the diffusion limit

$$\psi_i^{(2)}(a\epsilon) = \frac{1}{3}\psi_i^{(0)}(a\epsilon) \tag{182}$$

while in the free-streaming limit

$$\psi_i^{(2)}(a\epsilon) = \psi_i^{(0)}(a\epsilon) \tag{183}$$

We have chosen the following parameterization for the Eddington factor:

$$\mathcal{E}_i(\psi_i^{(0)}(a\epsilon), \psi_i^{(1)}(a\epsilon)) = \frac{1 + 2\mathcal{F}_{f\,i}^2(a\epsilon)}{3} \tag{184}$$

where the "flux-factor", $\mathcal{F}_{f\,i}$, is given by

$$\mathcal{F}_{f\,i}(a\epsilon) = \max\left[\mathrm{abs}\left(\frac{\psi_i^{(1)}(a\epsilon)}{\psi_i^{(0)}(a\epsilon)}\right), 1\right] \tag{185}$$

Neutrino Advection in Energy

Let us ow consider equation (163), the other part operator split in equation (155). This part determines the change in $\psi_i^{(0)}$ due to changes in a, b, and R. We begin by writing equation (163) can be written conservatively as

$$\left(\frac{\partial \Psi_i^{(0)}}{\partial \tau}\right)_{m,a\epsilon}^{(\epsilon)} + \frac{1}{(a\epsilon)^2}\left(\frac{\partial\Big((a\epsilon)^2 \mathcal{V}\Psi_i^{(0)}\Big)}{\partial a\epsilon}\right)_{t,m} = 0, \tag{186}$$

where

$$\Psi_i^{(0)}(a\epsilon) = \frac{bR^2}{a^3}\psi_i^{(0)}(a\epsilon), \tag{187}$$

and

$$\mathcal{V}(a\epsilon) = a\epsilon\left[\left(\frac{a_\tau}{a} - \frac{R_\tau}{R}\right) - \left(\frac{b_\tau}{b} - \frac{R_\tau}{R}\right)\mathcal{E}_i(a\epsilon)\right]. \tag{188}$$

and where $\mathcal{E}_i(a\epsilon)$ is the Eddington factor defined by above. Equation (186) is in the form of a continuity equation and describes the advection of neutrinos through the energy grid.

To difference equation (186) we use a PPM scheme that is second-order accurate in time and third-order accurate in energy, modeled after the advection scheme used in the PPM hydrodynamics code VH1. Briefly described, we are given the energy grid edges $(a\epsilon)^n_{j,k+\frac{1}{2}} \equiv a_j^n \epsilon^n_{k+\frac{1}{2}} = \epsilon_{\infty\,k+\frac{1}{2}}$ at the beginning of timestep n. The "velocities" of these grid edges, $\mathcal{V}^{n+\frac{1}{2}}_{k+\frac{1}{2}}$, are then computed, and a Lagrangian step is taken from time n to time $n+1$. In this step, the zone grid edges are moved in accordance with

$$(a\epsilon)^{n+1}_{j,k+\frac{1}{2}} = (a\epsilon)^n_{j,k+\frac{1}{2}} + \mathcal{V}^{n+\frac{1}{2}}_{j,k+\frac{1}{2}}\delta\tau_j^{n+\frac{1}{2}} \tag{189}$$

and the quantity $\Psi^{(0)}_{i\,j,k}$ is evolved by means of the equation

$$\begin{aligned}&\Psi^{(0)\,n+1}_{i\,j,k}\left[\left((a\epsilon)^{n+1}_{j,k+\frac{1}{2}}\right)^3 - \left((a\epsilon)^{n+1}_{j,k-\frac{1}{2}}\right)^3\right] \\ &= \Psi^{(0)\,n}_{i\,j,k}\left[\left((a\epsilon)^{n}_{j,k+\frac{1}{2}}\right)^3 - \left((a\epsilon)^{n}_{j,k-\frac{1}{2}}\right)^3\right]\end{aligned} \tag{190}$$

The energy grid edges are then remaped back to their original (time n) values and the $\Psi^{(0)\ n+1}_{i\ j+\frac{1}{2},k+\frac{1}{2}}$'s are advected through the zone edges using the PPM technology.

To test the accuracy of this scheme, we note that in the diffusion limit $\mathcal{E}_i(a\epsilon) = \frac{1}{3}$, and equation (188) for $\mathcal{V}$ becomes

$$\mathcal{V}(a\epsilon) = a\epsilon\left[\left(\frac{a_{,\tau}}{a} - \frac{R_{,\tau}}{R}\right) - \frac{1}{3}\left(\frac{b_{,\tau}}{b} - \frac{R_{,\tau}}{R}\right)\right] = a\epsilon\left(\frac{a_{,\tau}}{a} + \frac{1}{3}\frac{\rho_{,\tau}}{\rho}\right) \tag{191}$$

where we have used equations (121) and (43). Using equation (191) in equation (186) and expanding gives

$$\left(\frac{\partial\frac{\partial\mathcal{N}}{\partial(a\epsilon)}}{\partial\tau}\right)^{(\epsilon)}_{m,a\epsilon} + a\epsilon\left(\frac{a_{,\tau}}{a} + \frac{1}{3}\frac{\rho_{,\tau}}{\rho}\right)\left(\frac{\partial\left(\frac{\partial\mathcal{N}}{\partial(a\epsilon)}\right)}{\partial(a\epsilon)}\right)_{t,m} + \frac{\partial\mathcal{N}}{\partial(a\epsilon)}\left(\frac{\partial\left(a\epsilon\left(\frac{a_{,\tau}}{a} + \frac{1}{3}\frac{\rho_{,\tau}}{\rho}\right)\right)}{\partial(a\epsilon)}\right)_{t,m} = 0. \tag{192}$$

The characteristics of equation (192) are given by

$$\frac{d\tau}{1} = \frac{d(a\epsilon)}{a\epsilon\left(\frac{a_{,\tau}}{a} + \frac{1}{3}\frac{\rho_{,\tau}}{\rho}\right)} = -\frac{d\frac{\partial\mathcal{N}}{\partial(a\epsilon)}}{\frac{\partial\mathcal{N}}{\partial(a\epsilon)}\left(\frac{a_{,\tau}}{a} + \frac{1}{3}\frac{\rho_{,\tau}}{\rho}\right)} \tag{193}$$

From the first equality in equation (193), we get that

$$d\tau\left(\frac{a_{,\tau}}{a} + \frac{1}{3}\frac{\rho_{,\tau}}{\rho}\right) = \frac{d(a\epsilon)}{a\epsilon} \tag{194}$$

or

$$\frac{da}{a} + \frac{1}{3}\frac{d\rho}{\rho} - \frac{d(a\epsilon)}{a\epsilon} = 0. \tag{195}$$

Integrating equation (195), we get

$$\ln a + \frac{1}{3}\ln\rho - \ln a\epsilon = \text{constant} \tag{196}$$

which gives

$$a\rho^{1/3}(a\epsilon)^{-1} = \text{constant} \tag{197}$$

or

$$\epsilon\rho^{-1/3} = \text{constant} \tag{198}$$

Energy Equation

Consider now the energy equation (142). For matter in nuclear statistical equilibrium, we can regard the matter internal energy $e_{\rm m}$ as being a function of three thermodynamic variables. We will take these to be ρ, T, and $Y_{\rm e}$. Then (142) can be rewritten

$$\left(\frac{\partial e_{\rm m}}{\partial \rho}\right)_{T,Y_{\rm e}} \left(\frac{\partial \rho}{\partial \tau}\right)_m + \left(\frac{\partial e_{\rm m}}{\partial T}\right)_{\rho,Y_{\rm e}} \left(\frac{\partial T}{\partial \tau}\right)_m + \left(\frac{\partial e_{\rm m}}{\partial Y_{\rm e}}\right)_{\rho,T} \left(\frac{\partial Y_{\rm e}}{\partial \tau}\right)_m = -p_{\rm m} \left(\frac{\partial \left(\frac{1}{\rho}\right)}{\partial \tau}\right)_m - \frac{4\pi c^2}{(hc)^3} \int \epsilon^2 d\epsilon \sum_i {\rm RHS}_i^{(0)} \tag{199}$$

We will regard equation (195) as an equation for $\left(\frac{dT}{dt}\right)_m$ and operator split the contributions to $\left(\frac{dT}{d\tau}\right)_m$ in this equation as follows:

$$\left(\frac{dT}{d\tau}\right)_m = \left(\frac{dT}{d\tau}\right)_m^{(\rho)} + \left(\frac{dT}{d\tau}\right)_m^{(\nu)} \tag{200}$$

where $\left(\frac{dT}{d\tau}\right)_m^{(\rho)}$ arises from the hydrodynamics (i.e., changes in ρ) and $\left(\frac{dT}{d\tau}\right)_m^{(\nu)}$ arises from energy and lepton exchanges between matter and neutrinos. That is, $\left(\frac{dT}{d\tau}\right)_m^{(\rho)}$ and $\left(\frac{dT}{d\tau}\right)_m^{(\nu)}$ are given, respectively, by

$$\left(\frac{dT}{d\tau}\right)_m^{(\rho)} = -\frac{\left(\frac{\partial e_{\rm m}}{\partial \rho}\right)_{T,Y_{\rm e}} \left(\frac{\partial \rho}{\partial \tau}\right)_m + p_{\rm m} \left(\frac{\partial \left(\frac{1}{\rho}\right)}{\partial \tau}\right)_m}{\left(\frac{\partial e_{\rm m}}{\partial T}\right)_{\rho,Y_{\rm e}}} \tag{201}$$

and

$$\left(\frac{dT}{d\tau}\right)_m^{(\nu)} = -\frac{\left(\frac{\partial e_{\rm m}}{\partial Y_{\rm e}}\right)_{\rho,T} \left(\frac{\partial Y_{\rm e}}{\partial \tau}\right)_m + ac\frac{4\pi}{(hc)^3} \int \epsilon^2 d\epsilon \sum_\nu {\rm RHS}_i^{(0)}}{\left(\frac{\partial e_{\rm m}}{\partial T}\right)_{\rho,Y_{\rm e}}} \tag{202}$$

Equation (201) is differenced explicitly and solved iteratively to second order accuracy in time with the other hydro equations. Equation (198) is solved with the other MGFLD equations and is differenced implicitly as follows:

$$\left(\frac{\partial e_{\rm m}}{\partial T}\right)_{\rho,Y_{\rm e}}^{n+1} \frac{T_j^{n+1} - T_j^n}{\tau^{n+1} - \tau^n} + \left(\frac{\partial e_{\rm m}}{\partial Y_{\rm e}}\right)_{\rho,T}^{n+1} \frac{Y_{{\rm e}\,j}^{n+1} - Y_{{\rm e}\,j}^n}{\tau^{n+1} - \tau^n} - c\frac{4\pi}{(hc)^3} \sum_{k=1}^{N_{\rm E}} \epsilon_{j,k}^2 \Delta\epsilon_{j,k} \sum_i {\rm RHS}_{i\,j,k}^{(0)\,n+1} \tag{203}$$

Lepton Number Equation

The last of the set of MGFLD equations is the lepton continuity equation (143). This equation is solved with the other MGFLD equations and is differenced implicitly as follows:

$$\frac{Y_{\mathrm{e}\,j}^{n+1} - Y_{\mathrm{e}\,j}^{n}}{\tau^{n+1} - \tau^{n}} = -\frac{4\pi c}{(hc)^3}\,\frac{m_{\mathrm{B}}}{\rho^{n+1}} \sum_{k=1}^{N_{\mathrm{E}}} \epsilon_{j,k}^2 \Delta\epsilon_{j,k}^2 \times$$
$$\times \left[\left(j_{\nu_{\mathrm{e}}\,j,k}^{n+1} \left(1 - \psi_{\nu_{\mathrm{e}}\,j,}^{(0)\;n+1} \right) - \frac{\psi_{\nu_{\mathrm{e}}\,j,k}^{(0)\;n+1}}{\lambda_{\nu_{\mathrm{e}}\,j,k}^{n+1}} \right) - \left(j_{\bar{\nu}_{\mathrm{e}}\,j,k}^{n+1} \left(1 - \psi_{\bar{\nu}_{\mathrm{e}}\,j,k}^{(0)\;n+1} \right) - \frac{\psi_{\bar{\nu}_{\mathrm{e}}\,j,k}^{(0)\;n+1}}{\lambda_{\bar{\nu}_{\mathrm{e}}\,j,k}^{n+1}} \right) \right] \quad (204)$$

0.9.12 Solution of the MGFLD Equations

The form of the difference equations

Here we describe our numerical procedure for solving equations (162) and (170), the equations for the rate of change of $\psi_i^{(0)}$ due to sources and transport. along with equations (202) and (143), giving the associated changes in temperature and lepton number. Our difference equations corresponding to equations (162) and (170) are given by equations (164) and (165), and those corresponding to equations (202) and (143) are equations (203) and (204), respectively. Combining the left-hand side of equation (165) with (164), and differencing conservatively, gives

$$\frac{1}{c}\frac{\Delta\psi_{i\,j,k}^{(0)\;n+\frac{1}{2}}}{\Delta\tau^{n+\frac{1}{2}}} +$$
$$+(a_j)^2 \frac{-\frac{\mathrm{Area}_{j+\frac{1}{2}}}{(a_{j+\frac{1}{2}})^2}\frac{\lambda_{i\,j+\frac{1}{2},k}^{(t)\;n+1}}{3}\mathcal{F}_{i\,j+\frac{1}{2},k}^{n+1}\Gamma_{j+\frac{1}{2}}\frac{\psi_{i\,j+1,k}^{(0)\;n+1}-\psi_{i\,j,k}^{(0)\;n+1}}{\Delta R_{j+\frac{1}{2}}} + \frac{\mathrm{Area}_{j-\frac{1}{2}}}{(a_{j-\frac{1}{2}})^2}\frac{\lambda_{i\,j-\frac{1}{2},k}^{(t)\;n+1}}{3}\mathcal{F}_{i\,j-\frac{1}{2},k}^{n+1}\Gamma_{j-\frac{1}{2}}\frac{\psi_{i\,j,k}^{(0)\;n+1}-\psi_{i\,j-1,k}^{(0)\;n+1}}{\Delta R_{j-\frac{1}{2}}}}{\mathrm{Vol}_j}$$
$$= \mathrm{RHS}_i^{(0)\;n+1} \quad (205)$$

where

$$\Delta R_{j+\frac{1}{2}}^{n+1} = R_{j+1}^{n+1} - R_j^{n+1}, \quad R_j^{n+1} = \frac{R_{j+\frac{1}{2}}^{n+1} + R_{j-\frac{1}{2}}^{n+1}}{2} \quad (206)$$

$$\mathrm{Area}_{j+\frac{1}{2}} = 4\pi R_{j+\frac{1}{2}}^2, \quad \mathrm{Vol}_j = \frac{4\pi}{3}\left(R_{j+\frac{1}{2}}^3 - R_{j-\frac{1}{2}}^3 \right) \quad (207)$$

and

$$\psi_{i\,j+1,k}^{(0)\;n+1} = \psi_{i\,j+1,k}^{(0)\;n} + \Delta\psi_{i\,j+1,k}^{(0)\;n+\frac{1}{2}} \quad (208)$$

The quantities at time n are regarded as known, as are quantities without a superscripted time index. It is seen that the left-hand side of equation (205) for a given value of j

couples the unknowns $\Delta\psi^{(0)\ n+\frac{1}{2}}_{i\ j,k}$, $\Delta\psi^{(0)\ n+\frac{1}{2}}_{i\ j+1,k}$ and $\Delta\psi^{(0)\ n+\frac{1}{2}}_{i\ j-1,k}$, i.e., equation (205) couples the unknown at j with the unknowns at the two adjacent radial zones on either side of j.

The right-hand side of equation (162), namely the source term $\mathrm{RHS}^{(0)}_i$ for neutrinos of flavor i, has the form

$$
\begin{aligned}
\mathrm{RHS}^{(0)}_{i\ j,k} &= j^{(a)}_{i\ k}\left(1-\psi^{(0)}_{i\ j,k}\right) - \psi^{(0)}_{i\ j,k}/\lambda^{(a)}_{i\ k} \\
&+ \left[1-\psi^{(0)}_{i\ j,k}\right]\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\Phi^{\mathrm{in}}_{0\,\mathrm{LES\ i\ j,k,k'}}\psi^{(0)}_{i\ j,k'} - \psi^{(0)}_{i\ j,k}\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\Phi^{\mathrm{out}}_{0\,\mathrm{LES\ i\ j,k,k'}}\left[1-\psi^{(0)}_{i\ j,k'}\right] \\
&+ \left[1-\psi^{(0)}_{i\ j,k}\right]\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\langle\Phi^{\mathrm{in}}_{0\,\mathrm{SES\ i\ j,k,k'}}\rangle\psi^{(0)}_{i\ j,k'} - \psi^{(0)}_{i\ j,k}\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\langle\Phi^{\mathrm{out}}_{0\,\mathrm{SES\ i\ j,k,k'}}\rangle\left[1-\psi^{(0)}_{i\ j,k'}\right] \\
&+ \left[1-\psi^{(0)}_{i\ j,k}\right]\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\Phi^{\mathrm{p}}_{0\,\mathrm{TP\ i\ j,k,k'}}\left[1-\bar{\psi}^{(0)}_{i\ j,k'}\right] - \psi^{(0)}_{i\ j,k}\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\Phi^{\mathrm{a}}_{0\,\mathrm{LES\ i\ j,k,k'}}\bar{\psi}^{(0)}_{i\ j,k'} \\
&+ \left[1-\psi^{(0)}_{i\ j,k}\right]\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\sum_{k''=1}^{N_E}\epsilon^2_{k''}\Delta\epsilon_{k''}\sum_{i'}\Phi^{\mathrm{in}}_{0\,\nu\nu\mathrm{S\ i,j'\ j,k,k'}}\left\{\psi^{(0)}_{i\ j,k'}\psi^{(0)}_{i'\ j,k''}\left[1-\psi^{(0)}_{i'\ j,k'''}\right]\right\} \\
&-\psi^{(0)}_{i\ j,k}\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\sum_{k''=1}^{N_E}\epsilon^2_{k''}\Delta\epsilon_{k''}\sum_{i'}\Phi^{\mathrm{out}}_{0\,\nu\nu\mathrm{S\ i,j'\ j,k,k'}}\left\{\left[1-\psi^{(0)}_{i\ j,k'}\right]\left[1-\psi^{(0)}_{i'\ j,k''}\right]\psi^{(0)}_{i'\ j,k'''}\right\} \\
&+ \left[1-\psi^{(0)}_{i\ j,k}\right]\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\sum_{k''=1}^{N_E}\epsilon^2_{k''}\Delta\epsilon_{k''}\sum_{i'}\Phi^{\mathrm{p}}_{0\,\nu\nu\mathrm{TP\ i,j'\ j,k,k'}}\left\{\left[1-\bar{\psi}^{(0)}_{i\ j,k'}\right]\psi^{(0)}_{i'\ j,k''}\bar{\psi}^{(0)}_{i'\ j,k'''}\right\} \\
&-\psi^{(0)}_{i\ j,k}\sum_{k'=1}^{N_E}\epsilon^2_{k'}\Delta\epsilon_{k'}\sum_{k''=1}^{N_E}\epsilon^2_{k''}\Delta\epsilon_{k''}\sum_{i'}\Phi^{\mathrm{p}}_{0\,\nu\nu\mathrm{TP\ i,j'\ j,k,k'}}\left\{\bar{\psi}^{(0)}_{i\ j,k'}\left[1-\psi^{(0)}_{i'\ j,k''}\right]\left[1-\bar{\psi}^{(0)}_{i'\ j,k'''}\right]\right\}
\end{aligned}
\tag{209}
$$

The first line of equation (209) describes emission and absorption of neutrinos of flavor i, the second line describes the scattering of neutrinos of flavor i with energy transfers which in the mean are large in comparison with the energy bin width (e.g., neutrino–electron or neutrino–positron scattering), the third line describes the scattering of neutrinos of flavor i with energy transfers which in the mean are small in comparison with the energy bin width (e.g., neutrino–nucleon scattering), the fourth line describes the annihilation and production of neutrinos of flavor i and their antiparticles by electrons and positrons or by nucleon–nucleon bremsstrahlung, the fifth and sixth lines describe the in and out scattering, respectively, of neutrinos of flavor i by neutrinos of flavor i', and the seventh and eighth lines describe the production and annihilation, respectively, of neutrino-antineutrino pairs of flavor i by the annihilation and production neutrino pairs of flavor i'.

The first process, emission, absorption of neutrinos of flavor i does not couple energy zones. The second two processes, nonisoenergetic scattering from non-neutrino particles, couples energy zones at each given pair, i, j, of flavor and radial zone indices. The fourth process, annihilation and production of neutrinos and antineutrinos of flavor

i by electrons and positrons or by nucleon–nucleon bremsstrahlung, couples the energy zones of both neutrinos of flavor i and their antiparticles at each given radial index j. Finally, the last two processes, the scattering of neutrinos of flavor i by neutrinos of flavor i', and the production and annihilation of neutrino-antineutrino pairs of flavor i by the annihilation and production neutrino pairs of flavor i', couples all the energy zones of all the neutrino flavors for each radial index j.

In order that all of the above source terms be included in CHIMERA it is essential for numerical stability that a solution for all the unknowns $\Delta\psi_{i\ j+1,k}^{(0)\ n+\frac{1}{2}}$ along with the ΔT_j's and $\Delta Y_{\mathrm{e}\ j}$'s be carried out implicitly and simultaneously. In CHIMERA the ν_μ's and ν_τ's (collectively ν_{x}'s) are treated as one neutrino flavor with a degeneracy factor of 2, as are the $\bar{\nu}_\mu$'s and $\bar{\nu}_\tau$'s (collectively $\bar{\nu}_{\mathrm{x}}$'s). This is because none of these neutrino flavor groups are able to undergo charged current reactions for kinematical reasons, and the neutral current reactions for the two flavors within each group are identical. Thus, CHIMERA separately evolves four neutrino flavors, viz., ν_{e}'s, $\bar{\nu}_{\mathrm{e}}$'s, ν_{x}'s (ν_μ's and ν_τ's), and $\bar{\nu}_{\mathrm{x}}$'s ($\bar{\nu}_\mu$'s and $\bar{\nu}_\tau$'s).

Linearization of the difference equations

To solve for the unknowns $\Delta\psi_{i\ j+1,k}^{(0)\ n+\frac{1}{2}}$, $\Delta T_j^{n+\frac{1}{2}}$, and $\Delta Y_{\mathrm{e}\ j}^{n+\frac{1}{2}}$, equations (204) and (209) are linearized and a solution of the resulting coupled linear equations for the unknowns $\Delta\psi_{i\ j+1,k}^{(0)\ n+\frac{1}{2}}$ and $\Delta Y_{\mathrm{e}\ j}$ are obtained by a linear equation solver. These solutions are regarded as corrections to the $\psi_{i\ j+1,k}^{(0)\ n}$'s and $Y_{\mathrm{e}\ j}^{n}$'s and added to them, and the process is repeated until no further relative change of any of the $\Delta\psi_{i\ j+1,k}^{(0)\ n+\frac{1}{2}}$'s and $\Delta Y_{\mathrm{e}\ j}$'s occurs that exceeds a small prescribed tolerance, which is taken to be 10^{-6} in CHIMERA. After each iteration cycle the corrections $\Delta T_j^{n+\frac{1}{2}}$ to the temperatures are solved by using equation (203). The reason for taking the $\Delta T_j^{n+\frac{1}{2}}$'s out of the direct implicit linear solve is that the specific heat of the matter is typically large in comparison with that of the neutrinos, so the $\Delta T_j^{n+\frac{1}{2}}$'s tend to be small and their removal from the implicit linear solve does not impair convergence. The advantage of removing them from the implicit linear solve is that the conditioning of the linear equations is thereby improved by many orders of magnitude.

Lineariizing equation equation (204), omitting the last two sources in equation (209) that are not yet in CHIMERA, collapsing the expressions for neutrino scattering with large and with small energy exchanges into one expression, and writing it for the $i^{\mathrm{th}}+1$ iteration in going from time step n to time step $n+1$, denoting quantities to be evaluated at time step $n+1$ and known through iteration i by a superscript $n+1,\ it$ (e.g., $X^{n+1,\ it}$), quantities straddling time step n and time step $n+1$ and known through iteration i by a superscript $n+\frac{1}{2},\ it$(e.g., $\Delta X^{n+\frac{1}{2},\ it}$), and the unknown corrections straddling time step n and $n+1$ and being solved for at the current $it+1$ iteration by superscript $n+\frac{1}{2},\ it+1$ (e.g., $\delta X^{n+\frac{1}{2},\ it+1}$). we have

$$\frac{1}{c}\frac{\Delta\psi^{(0)\ n+\frac{1}{2},\ it}_{i\ j,k}}{\Delta\tau^{n+\frac{1}{2}}}+\frac{1}{\mathrm{vol}_j}\left[\mathrm{area}_{j+\frac{1}{2}}\psi^{(1)\ n+1,\ it}_{i\ j+\frac{1}{2},k}-\mathrm{area}_{j-\frac{1}{2}}\psi^{(1)\ n+1,\ it}_{i\ j-\frac{1}{2},k}\right]$$

$$+\frac{\mathrm{area}_{j+\frac{1}{2}}}{\mathrm{vol}_j}\left(-\mathcal{D}^{n+1,\ it}_{i\ j+\frac{1}{2},k}\frac{\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j+1,k}}{\Delta r_{j+\frac{1}{2}}}\right)+\frac{\mathrm{area}_{j+\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j+\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j+\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial\psi^{(0)}_{i\ j+1,k}}\right)^{n+\frac{1}{2},\ it}_{j+\frac{1}{2},k}\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j+1,k}$$

$$+\frac{-\mathrm{area}_{j-\frac{1}{2}}}{\mathrm{vol}_j}\left(-\mathcal{D}^{n+1,\ it}_{i\ j-\frac{1}{2},k}\frac{-\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j-1,k}}{\Delta r_{j-\frac{1}{2}}}\right)+\frac{-\mathrm{area}_{j-\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j-\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j-\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial\psi^{(0)}_{i\ j-1,k}}\right)^{n+\frac{1}{2},\ it}_{j-\frac{1}{2},k}\delta\psi^{(0)\ n+\frac{1}{2},\ i}_{i\ j-1,k}$$

$$+\frac{1}{c}\frac{\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}}{\Delta\tau^{n+\frac{1}{2}}}+$$

$$+\frac{1}{\mathrm{vol}_j}\left[\mathrm{area}_{j+\frac{1}{2}}\left(-\mathcal{D}^{n+1,\ it}_{i\ j+\frac{1}{2},k}\frac{-\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}}{\Delta r_{j+\frac{1}{2}}}\right)-\mathrm{area}_{j-\frac{1}{2}}\left(-\mathcal{D}^{n+1,\ it}_{i\ j-\frac{1}{2},k}\frac{\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}}{\Delta r_{j-\frac{1}{2}}}\right)\right]$$

$$+\frac{\mathrm{area}_{j+\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j+\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j+\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial\psi^{(0)}_{i\ j,k}}\right)^{n+\frac{1}{2},\ it}_{j+\frac{1}{2},k}\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}-\frac{\mathrm{area}_{j-\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j-\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j-\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial\psi^{(0)}_{i\ j,k}}\right)^{n+\frac{1}{2},\ it}_{j-\frac{1}{2},k}\delta\psi^{(0)\ n+}_{i\ j,k}$$

$$+\frac{\mathrm{area}_{j+\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j+\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j+\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial Y_{\mathrm{e}\ j+1}}\right)^{n+\frac{1}{2},\ it}_{j+\frac{1}{2},k}\delta Y^{n+\frac{1}{2},\ it+1}_{\mathrm{e\ j+1}}-\frac{\mathrm{area}_{j-\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j-\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j-\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial Y_{\mathrm{e}\ j-1}}\right)^{n+\frac{1}{2},\ it}_{j-\frac{1}{2},k}\delta Y^{n+\frac{1}{2},}_{\mathrm{e\ j-1}}$$

$$+\frac{\mathrm{area}_{j+\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j+\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j+\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial Y_{\mathrm{e}\ j}}\right)^{n+\frac{1}{2},\ it}_{j+\frac{1}{2},k}\delta Y^{n+\frac{1}{2},\ it+1}_{\mathrm{e\ j}}-\frac{\mathrm{area}_{j-\frac{1}{2}}}{\mathrm{vol}_j}\frac{\psi^{(1)\ n+1,\ it}_{i\ j-\frac{1}{2},k}}{\mathcal{D}^{n+1,\ it}_{i\ j-\frac{1}{2},k}}\left(\frac{\partial\mathcal{D}_i}{\partial Y_{\mathrm{e}\ j}}\right)^{n+\frac{1}{2},\ it}_{j-\frac{1}{2},k}\delta Y^{n+\frac{1}{2},\ it+}_{\mathrm{e\ j}}$$

$$=j^{n+1,\ it}_{i\ j,k}\left(1-\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)-\left(\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{1}{\lambda^{(a)\ n+1,\ it}_{i\ j,k}}$$

$$+\left(1-\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}\epsilon_{k'}\left(\Phi^{(\mathrm{in})\ n+1,\ it}_{0\ \mathrm{NIS}\ i\ j,k.k'}\right)\left(\psi^{(0)\ n+1,\ it}_{i\ j,k'}\right)$$

$$-\left(\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\left(\Phi^{(\mathrm{out})\ n+1,\ it}_{0\ \mathrm{NIS}\ i\ j,k.k'}\right)\left(1-\psi^{(0)\ n+1,\ it}_{i\ j,k'}\right)$$

$$+\left(1-\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\left(\Phi^{(\mathrm{p})\ n+1,\ it}_{0\ \mathrm{TP}\ i\ j,k.k'}\right)\left(1-\bar{\psi}^{(0)\ n+1,\ it}_{i\ j,k'}\right)$$

$$-\left(\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\Phi^{(\mathrm{a})\ n+1}_{0\ \mathrm{TP}\ i\ j,k.k'}\left(\bar{\psi}^{(0)\ n+1,\ it}_{i\ j,k'}\right)$$

$$-\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}\left(j^{n+1\ it}_{i\ j,k}+\left(\frac{1}{\lambda^{(a)}_{i\ j,k}}\right)^{n+1\ it}_{j}\right),$$

$$-\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\left(\Phi^{(\mathrm{in})\ n+1,\ it}_{0\ \mathrm{NIS}\ i\ j,k.k'}\right)\left(\psi^{(0)\ n+1,\ it}_{i\ j,k'}\right)$$

$$-\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k}\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\left(\Phi^{(\mathrm{out})\ n+1,\ it}_{0\ \mathrm{NIS}\ i\ j,k.k'}\right)\left(1-\psi^{(0)\ n+1,\ it}_{i\ j,k'}\right)$$

$$+\left(1-\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\left(\Phi^{(\mathrm{in})\ n+1,\ it}_{0\ \mathrm{NIS}\ i\ j,k.k'}\right)\left(\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k'}\right)$$

$$-\left(\psi^{(0)\ n+1,\ it}_{i\ j,k}\right)\frac{2\pi}{(2\pi\hbar c)^3}\sum_{k'=1}^{N_{\mathrm{E}}}\epsilon^2_{k'}d\epsilon_{k'}\left(\Phi^{(\mathrm{out})\ n+1,\ it}_{0\ \mathrm{NIS}\ i\ j,k.k'}\right)\left(-\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ j,k'}\right)$$

where

$$\mathcal{D}^{n+1,\,it}_{i\;j+\frac{1}{2},k} = \frac{(a_j)^2}{(a_{j+\frac{1}{2}})^2}\frac{\lambda^{(t)\;n+1,\,it}_{i\;j+\frac{1}{2},k}}{3}\mathcal{F}^{n+1,\,it}_{i\;j+\frac{1}{2},k}\Gamma_{j+\frac{1}{2}} \tag{211}$$

Linearizing equation (209) gives

$$\begin{aligned}
&\frac{1}{c}\frac{\Delta Y^{(0)\;n+1\;it}_{\mathrm{e}\;j}}{\Delta\tau^{n+1/2}} + \frac{1}{c}\frac{\delta Y^{(0)\;n+1\;it+1}_{\mathrm{e}\;j}}{\Delta\tau^{n+1/2}} \\
&= -\frac{4\pi}{(hc)^3}\frac{m_{\mathrm{B}}}{\rho_j}\sum_{k=1}^{N_{\mathrm{E}}}\epsilon_k^2 d\epsilon_k \times \left[j^{n+1\;it}_{\nu_{\mathrm{e}},j,k}\left(1-\psi^{(0)\;n+1,\;it}_{\nu_{\mathrm{e}},j,k}\right) - \psi^{(0)\;n+1,\;it}_{\nu_{\mathrm{e}},j,k}\frac{1}{\lambda^{(a)\;n+1\;it}_{\nu_{\mathrm{e}}\;j,k}}\right] \\
&+\frac{4\pi}{(hc)^3}\frac{m_{\mathrm{B}}}{\rho_j}\sum_{k=1}^{N_{\mathrm{E}}}\epsilon_k^2 d\epsilon_k \times \left[j^{n+1\;it}_{\bar{\nu}_{\mathrm{e}},j,k}\left(1-\psi^{(0)\;n+1,\;it}_{\bar{\nu}_{\mathrm{e}},j,k}\right) - \psi^{(0)\;n+1,\;it}_{\bar{\nu}_{\mathrm{e}},j,k}\frac{1}{\lambda^{(a)\;n+1\;it}_{\bar{\nu}_{\mathrm{e}}\;j,k}}\right] \\
&-\frac{4\pi}{(hc)^3}\frac{m_{\mathrm{B}}}{\rho_{j-1/2}}\sum_{k=1}^{N_{\mathrm{E}}}\epsilon_k^2 d\epsilon_k \times \\
&\times\left[j^{n+1\;it}_{\nu_{\mathrm{e}},j,k}\left(-\delta\psi^{(0)\;n+\frac{1}{2},\;it+1}_{\nu_{\mathrm{e}},j,k}\right) - \delta\psi^{(0)\;n+\frac{1}{2},\;it+1}_{\nu_{\mathrm{e}},j,k}\frac{1}{\lambda^{(a)\;n+1\;it}_{\nu_{\mathrm{e}}\;j,k}}\right. \\
&+\left(\frac{\partial j_{\nu_{\mathrm{e}}}}{\partial Y_{\mathrm{e}}}\right)^{n+\frac{1}{2},\;it}_{j,k}\delta Y^{n+\frac{1}{2},\;it+1}_{\mathrm{e\;j}}\left(1-\psi^{(0)\;n+1,\;it}_{\nu_{\mathrm{e}},j,k}\right) \\
&-\psi^{(0)\;n+1,\;it}_{\nu_{\mathrm{e}},j,k}\left(\frac{\partial(1/\lambda^{(a)}_{\nu_{\mathrm{e}}})}{\partial Y_{\mathrm{e}}}\right)^{n+\frac{1}{2},\;it}_{j,k}\delta Y^{n+\frac{1}{2},\;it+1}_{\mathrm{e\;j}} \\
&-j^{n+1\;it}_{\bar{\nu}_{\mathrm{e}},j,k}\left(-\delta\psi^{(0)\;n+\frac{1}{2},\;it+1}_{\bar{\nu}_{\mathrm{e}},j,k}\right) - \delta\psi^{(0)\;n+\frac{1}{2},\;it+1}_{\bar{\nu}_{\mathrm{e}},j,k}\frac{1}{\lambda^{(a)\;n+1\;it}_{\bar{\nu}_{\mathrm{e}}\;j,k}} \\
&-\left(\frac{\partial j_{\bar{\nu}_{\mathrm{e}}}}{\partial Y_{\mathrm{e}}}\right)^{n+\frac{1}{2},\;it}_{j,k}\delta Y^{n+\frac{1}{2},\;it+1}_{\mathrm{e\;j}}\left(1-\psi^{(0)\;n+1,\;it}_{\bar{\nu}_{\mathrm{e}},j,k}\right) \\
&\left.+\;\psi^{(0)\;n+1,\;it}_{\bar{\nu}_{\mathrm{e}},j,k}\left(\frac{\partial(1/\lambda^{(a)}_{\bar{\nu}_{\mathrm{e}}})}{\partial Y_{\mathrm{e}}}\right)^{n+\frac{1}{2},\;it}_{j,k}\delta Y^{n+\frac{1}{2},\;it+1}_{\mathrm{e\;j}}\right]
\end{aligned} \tag{212}$$

Solution of the linearized difference equations

Equations (210) and (212) can be written in the following compact form

$$\sum_{k'=0}^{4N_E+1}\left(A^{i-1}_{j,k,k'}u^i_{j,k'}\right) + D^{(-)\,i-1}_{j,k}u^i_{j-1,k} + D^{(+)\,i-1}_{j,k}u^i_{j+1,k} + C^{i-1}_{j,k} = 0 \tag{213}$$

where for a given j

$$u_{j,k} = \begin{pmatrix} \delta\psi_{1\ j,1}^{(0)\ n+\frac{1}{2},\ it+1} \\ \vdots \\ \delta\psi_{1\ j,N_E}^{(0)\ n+\frac{1}{2},\ it+1} \\ \delta\psi_{2\ j,1}^{(0)\ n+\frac{1}{2},\ it+1} \\ \vdots \\ \delta\psi_{2\ j,N_E}^{(0)\ n+\frac{1}{2},\ it+1} \\ \delta\psi_{3\ j,1}^{(0)\ n+\frac{1}{2},\ it+1} \\ \vdots \\ \delta\psi_{3\ j,N_E}^{(0)\ n+\frac{1}{2},\ it+1} \\ \delta\psi_{4\ j,1}^{(0)\ n+\frac{1}{2},\ it+1} \\ \vdots \\ \delta\psi_{4\ j,N_E}^{(0)\ n+\frac{1}{2},\ it+1} \\ \delta Y_{\mathrm{e\ j}}^{n+\frac{1}{2},\ it+1} \end{pmatrix} \tag{214}$$

and where we now extend the k and k' index to run over both the four values of i, the N_E values of k and k' and one additional value pertaining to the $\delta Y_{\mathrm{e\ j}}^{n+\frac{1}{2},\ it+1}$ variable, $4N_E+1$ values in all. For each value of j there are thus $4N_E+1$ equations containing the $4N_E+1$ unknowns at j, and $4N_E$ unknowns both at $j-1$ and $j+1$. (The $D_{j,k}^{(-)\ i-1}$'s and $D_{j,k}^{(+)\ i-1}$'s are zero for $k = 4N_E+1$.)

A boundary condition at the inner edge ($j = 1$) and at the outer edge ($j = N_J$) are clearly required to complete this set of equations.

Suppose that the boundary condition at the center can be represented by

$$u_{0,k} = \sum_{k'=0}^{4N+1} \left(R_{0,k.k'} u_{1,k'}^{i} \right) + S_{0,k} \tag{215}$$

Using equation (215) in (213) for $j = 1$, gives

$$
\begin{aligned}
0 = \sum_{k'=0}^{4N+1} (A_{1,k,k'} u_{1,k'}) + D_{1,k}^{(-)} \left[\sum_{k'=0}^{4N+1} (R_{0,k,k'} u_{1,k'}) + S_{0,k} \right] \\
+ D_{1,k}^{(+)} u_{2,k} + C_{1,k} \\
= \sum_{k'=0}^{4N+1} (A_{1,k,k'} u_{1,k'}) + D_{k.j}^{(-)} \sum_{k'=0}^{4N+1} (R_{0,k.k'} u_{1,k'}) + D_{1,k}^{(-)} S_{0,k} \\
+ D_{1,k}^{(+)} u_{2,k} + C_{1,k} \\
= \sum_{k'=0}^{4N+1} \left(A_{1,k,k'} + D_{1,k}^{(-)} R_{0,k.k'} \right) u_{1,k'} + D_{1,k}^{(-)} S_{0,k} + C_{1,k} \\
+ D_{1,k}^{(+)} u_{2,k}
\end{aligned}
\tag{216}
$$

In CHIMERA we impose symmetric boundary conditions at the center, i.e.,

$$
u_{0,k} = u_{1,k} \tag{217}
$$

thus eliminating any reference to the unknowns $u_{0,k}$. Reference to equation (215) shows that for these boundary conditions will be satisfied if

$$
R_{0,k.k'} = \delta_{k,k'} \qquad S_{0,k} = 0. \tag{218}
$$

At the outer edge we impose boundary conditions of the form

$$
u_{N_J+1,k} = \mathcal{R}_k \times u_{N_J,k}. \tag{219}
$$

$\mathcal{R}_k$ is obtained from the equation for $\psi^{(1)}_{i\ N_J+\frac{1}{2},k}$:

$$
\psi^{(1)\,n+1}_{i\ N_J+\frac{1}{2},k} = -\frac{\lambda^{(t)\,n+1}_{i\ N_J+\frac{1}{2},k}}{3} \mathcal{F}^{n+1}_{i\ N_J+\frac{1}{2},k} \Gamma^{n+1}_{N_J+\frac{1}{2}} \frac{\psi^{(0)\,n+1}_{i\ N_J+1,k} - \psi^{(0)\,n+1}_{i\ N_J,k}}{R^{n+1}_{N_J+1} - R^{n+1}_{N_J}}, \tag{220}
$$

and the free streaming condition

$$
\psi^{(1)\,n+1}_{i\ N_J+\frac{1}{2},k} = \frac{\psi^{(0)\,n+1}_{i\ N_J+1,k} + \psi^{(0)\,n+1}_{i\ N_J,k}}{2} \tag{221}
$$

Elimination of $\psi^{(1)\,n+1}_{i\ N_J+\frac{1}{2},k}$ between equations (220) and (221) allows $\psi^{(0)\,n+1}_{i\ N_J+1,k}$ to be expressed in terms of $\psi^{(0)\,n+1}_{i\ N_J,k}$, and therefore $\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ N_J+1,k}$ in terms of $\delta\psi^{(0)\ n+\frac{1}{2},\ it+1}_{i\ N_J,k}$.

Our solution for the corrections at each iteration thus proceeds as follows. Linear equations (216) are solved for the $u_{1,k}$ in terms of the $u_{2,k}$. Using this solution for the $u_{1,k}$'s in equations (213) with $j = 2$, these equations now involve only the the $u_{2,k}$'s and the $u_{3,k}$'s and are then solved for the $u_{2,k}$'s in terms of the $u_{3,k}$'s. This procedure in repeated for increasing values of j until $j = N_J$, at which point the $u_{N_J,k}$'s are

solved in terms of the $u_{N_J+1,k}$'s. The outer boundary conditions (218) now permit the solution for the $u_{N_J,k}$'s to be obtained. Using the solution for the $u_{N_J,k}$'s in the equations for the $u_{N_J-1,k}$'s in terms of the $u_{N_J,k}$'s obtained previously permits the solution for the $u_{N_J-1,k}$'s to be now obtained. Proceeding to successively smaller values of j until $j = 1$ is reached, we obtain the solution for all the $u_{j,k}$'s.

0.10 CHIMERA's Network and Equation of State Modules

The composition and state of the stellar matter, largely electrons, photons and atomic nuclei, has profound impact on the hydrodynamic evolution and neutrino radiation transport in core collapse supernovae. For example, the matter pressure, which depends on the composition and state of the matter as well as its temperature and density, drives much of the hydrodynamic motion. Similarly, the composition of the matter has a great effect on the neutrino opacities, determining how effectively energy is transfered from the neutrino radiation field to the matter. In addition, the composition of the ejecta, rich in newly synthesized heavy elements, is one of the key outcomes of the supernova explosion and intimately tied to many of the potential observations of these events. For these reasons, considerable effort is invested in CHIMERA in evolving the composition and calculating the resulting equation of state.

The center of a massive star as it nears its demise is composed of iron, nickel, and similar elements (collectively called the iron-peak nuclei), the end products of stellar nucleosynthesis. The maximally bound, iron-peak nuclei are the result of thermodynamic equilibrium for strong and electromagnetic nuclear reactions, termed *Nuclear Statistical Equilibrium* (NSE). Above this *iron core* lie concentric layers of successively lighter elements, recapitulating the sequence of nuclear burning that occurred in the core during the star's lifetime. Once the iron core grows too massive to be supported by electron degeneracy pressure, core collapses ensues. During collapse, the increasing density and neutronization causes NSE to favor heavier, more neutron-rich nuclei. This leads eventually to a composition dominated by exotic columnar and planar nuclear states [135] and ultimately, when the core reaches densities similar to those of the nucleons in a nucleus, nuclear matter.

To accurately model the evolution of the composition in a supernova, a combination of methods must be used to cover regions where NSE does not occur as well as regions where NSE results in a composition dominated by nuclei like those found on Earth, more exotic nuclear states or nuclear matter. We will detail these methods in the following sub-sections. In addition to the nuclear composition, the populations of electrons and positrons must also be tracked and the photon population is assumed to obey local thermodynamic equilibrium. In CHIMERA, the impact of the leptonic contribution is included via an electron-positron EOS with arbitrary degeneracy and degree of relativity that spans the entire density and temperature regime of interest. This EOS component is based on the electron-positron component of [45], but has been extended into the non-relativistic regime by Bruenn. This EOS also includes the contribution of the thermal photonic radiation field.

0.10.1 Nuclear Statistical Equilibrium

At conditions of high temperature and density, thermonuclear reaction rates may be sufficiently rapid to achieve equilibrium within the timescale set by the hydrodynamics of the astrophysical setting. This permits considerable simplification of the calculation of the nuclear abundances. In most such cases, the fast strong and electromagnetic reactions reach equilibrium while those involving the weak nuclear force do not. Since the weak reactions are not equilibrated, the resulting Nuclear Statistical Equilibrium requires monitoring of weak reaction activity, including the effects of neutrino interactions. Even with this stricture, NSE offers many advantages, since hundreds of abundances are uniquely defined by the thermodynamic conditions and a single measure of the weak interaction history or the degree of neutronization. Computationally, this reduction in the number of independent variables greatly reduces the cost of nuclear abundance evolution. Because there are fewer variables to follow within a hydrodynamic model, the memory footprint of the nuclear abundances is also reduced, an issue of importance in modern multi-dimensional models. Finally, the equilibrium abundance calculations depend on binding energies and partition functions, quantities which are better known than many reaction rates. This is particularly true for unstable nuclei and for conditions where the mass density approaches that of the nucleus itself, resulting in exotic nuclear structures.

The expression for NSE is commonly derived using either chemical potentials or detailed balance [43]. A nucleus with atomic number Z and atomic mass A, denoted ${}^{A}Z$, is composed of Z protons and $N = (A - Z)$ neutrons. In astrophysics, it is common to discuss the composition of isotopic species i in terms of its (molar) abundance, Y_i, which can be derived from the number density, n_i, via the relation $Y_i = n_i/\rho N_A$, where N_A is Avogadro's number and ρ is the matter mass density. Y_i has units of molg^{-1}. Another useful descriptor of the composition in the mass fraction, X_i, representing the fractional content by mass of species i in the matter. For a nucleus with atomic weight A_i, $X_i = A_i Y_i$. X_i is dimensionless as A_i has units of gmol^{-1}. In equilibrium with these free nucleons, the chemical potential of ${}^{A}Z$ can be expressed in terms of the chemical potentials of the free nucleons

$$\mu_{Z,A} = Z\mu_p + N\mu_n \ . \tag{222}$$

Substituting the expression for the Boltzmann chemical potential (including rest mass) into Eq. 222 allows derivation of an expression for the abundance of every nuclear species in terms of the abundances of the free protons (Y_p) and neutrons (Y_n),

$$Y({}^{A}Z) = \frac{G({}^{A}Z)}{2^A}\left(\frac{\rho N_A}{\theta}\right)^{A-1} A^{\frac{3}{2}} \exp\left(\frac{B({}^{A}Z)}{k_B T}\right) {Y_n}^{N} {Y_p}^{Z}, \tag{223}$$

where $G({}^{A}Z)$ and $B({}^{A}Z)$ are the partition function and binding energy of the nucleus ${}^{A}Z$, N_A is Avagadro's number, k_B is Boltzmann's constant, ρ and T are the density and temperature of the plasma, and $\theta = (m_u k_B T/2\pi\hbar^2)^{3/2}$.

For a given set of nuclei whose binding energies and partition functions are known, abundances of all nuclear species can therefore be expressed as functions of two quantities. Nucleon number conservation ($\sum AY = 1$) provides one constraint. The second

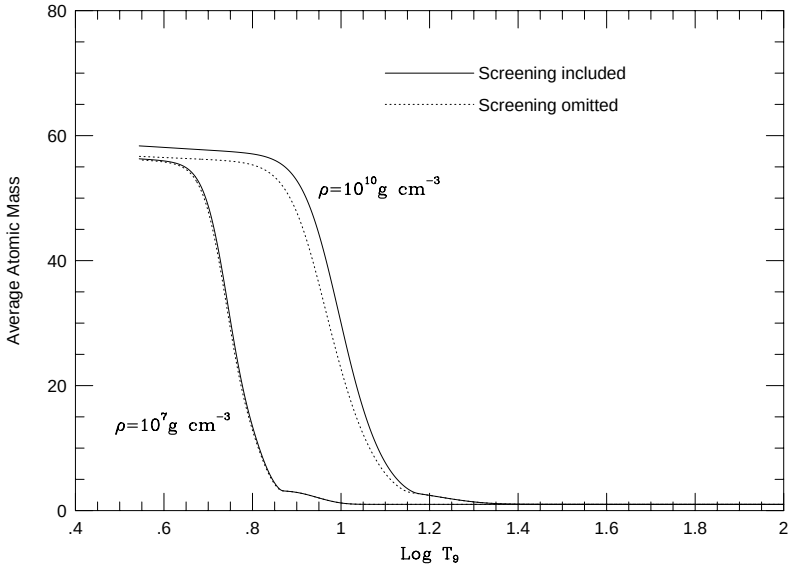


Figure 2: The average atomic mass for material in NSE as a function of Temperature. The solid lines include Coloumb (screening) corrections to the nuclear binding energies, while the dotted lines ignore this effect.

constraint is the amount of weak reaction activity, often expressed in terms of the total proton abundance, $\sum ZY$, which charge conservation requires equal the electron abundance, . Thus the nuclear abundances are uniquely determined for a given $(T, \rho,)$. Alternately, the weak interaction history is sometimes expressed in terms of the neutron excess $\eta = \sum (N - Z)Y$. Figure 2 displays the temperature and density dependence of $\bar{A} = \sum AY / \sum Y = 1 / \sum Y$, the average nuclear mass of the NSE distribution. At high temperatures, free nucleons are favored, hence $\bar{A} \sim 1$. For intermediate temperatures the compromise of retaining large numbers of particles while increasing binding energy favors ^{4}He, which has 80% of the binding energy of the iron peak nuclei. At low temperatures, Eq. 223 strongly favors the most bound nuclei, the iron peak nuclei, so $\bar{A} \rightarrow 60$ as the temperature drops. Density can be seen to scale the placement of these divisions between high, intermediate and low temperature. Increasing density, through the $\rho^{(A-1)}$ dependency, can be seen to increase the average mass at low temperature. Variations in neutronization do not strongly affect Figure 2. At high temperatures, it simply effects the ratio of $Y_p / Y_n \approx /1-$. At low temperatures, variation in changes which Fe-peak isotopes dominate. For example, though ^{5}6Ni is less tightly bound than ^{5}4Fe, it is more tightly bound than ^{5}4Fe + 2 ^{1}H, which would be required by charge conservation if $\sim .5$. Thus $Y(^{56}Ni) > Y(^{54}Fe)$ for low T with $\sim .50$,

but $Y(^{54}Fe) > Y(^{56}Ni)$ for smaller . In general, the most abundant nuclei at low temperatures are the most bound nuclei for which $Z/A \sim$.

As with any equilibrium distribution, there are limitations on the applicability of NSE. For NSE to provide a good estimate of the nuclear abundances the temperature must be sufficient for the endoergic reaction of each reaction pair to occur. Since for all particle-stable nuclei between the proton and neutron drip lines (with the exception of nuclei unstable against alpha decay), the photodisintegrations are endoergic, with typical Q-values among (β) stable nuclei of 8-12 , by Eq. 230 this requirement reduces to $T > 3$. While this requirement is necessary, it is not sufficient. In the case of hydrostatic silicon burning, even when this condition is met, appreciable time (hours to days) is required to convert Si to Fe-peak elements. In the case of explosive silicon burning, the adiabatic cooling on timescales of seconds can cause conditions to change more rapidly than NSE can follow, breaking down NSE first between ^{4}He and ^{12}C, at $T \sim 6$ [103] and later between the species near silicon and the Fe-peak nuclei, at $T \sim 4$ [156, 74]. Thus it is clear that in the face of sufficiently rapid thermodynamic variations, NSE provides a problematic estimate of abundances so a transition to a reaction network is necessary both for matter which has not yet reached NSE and for matter which had been in NSE but is cooling.

In CHIMERA v1.0, Nuclear Statistical Equilibrium (NSE) is assumed in regions where the temperature exceeds 5.5 GK or where matter reached NSE during stellar evolution (the initial iron core). For densities less than $1.7 \times 10^{8} \frac{g}{cm^3}$, matter in NSE is described by a highly modified version of the EOS described by [45]. Under these conditions the Cooperstein EOS assumes NSE between of free neutrons, free protons, α-particles, and a representative or average heavy nucleus, whose charge number and mass adjust to the changing conditions. This representative nucleus approach greatly lightens the computational load of calculating NSE and has been shown to be sufficient for calculations of equation of state quantities such as pressure [34]. It is however insufficient for neutrino interactions, as the neutrino interactions rates sample the shell structure of nuclei and therefore may vary by orders of magnitude between neighboring nuclei. Thus the average rate of a neutrino capture, for example, is not well represented by the capture rate on the average nucleus. For purposes of neutrino opacities, we therefore adopt a more detailed NSE using mass and partition function data from [131] to build tabulations of the needed interaction rates. For CHIMERA v2.0, we anticipate adopting a similar scheme as a replacement for the Cooperstein EOS to provide a better match with the larger reaction network we will use in v2.

0.10.2 The High Density Equation of State and Nuclear Matter

Many of the nuclei present in the iron core at the onset of collapse are stable or long-lived, thus their properties are well known, having been studied in terrestrial laboratories. However as core collapse proceeds, the combined effects of increasing density and the resulting electron capture drives the composition toward heavier, more-neutron rich nuclei. Knowledge of the nuclear properties needed to determine the composition of the matter and its equation of state under these conditions is based on extrapolations from laboratory measurements that become increasingly suspect as density increases. These extrapolations breakdown completely at high densities, particularly as columnar

or planar nuclei and nuclear matter become possibilities. Because the density of nucleons outside of the nucleus becomes important as high densities, a consistent treatment nucleons and nuclei, including modifications to the nuclear surface, becomes important as does consideration of the finite volume of the nucleons and their Fermi-Dirac nature. Additionally, consideration must be paid to the phase transition between nuclear and nucleonic phases. For these reasons, NSE calculated as in Eq. 223 is insufficient at high density.

To address these conditions, CHIMERA uses the equation of state (EOS) of [91] for regions in NSE where the density exceeds $1.7 \times 10^8 \frac{g}{cm^3}$. This EOS is a more computational efficient approximation to the EOS of [88, 92], using unit cell calculations for a compressible liquid drop model of both finite nuclei and nuclear matter. Like the Cooperstein EOS, the LS EOS assumes NSE between of free neutrons, free protons, α-particles, and a representative or average heavy nucleus, whose charge number and mass adjust to the changing conditions. The LS EOS has been the industry standard EOS for supernova simulations for more than a decade. With LS EOS as a baseline, we plan to explore with CHIMERA the impact of the nuclear equation of state on the supernova mechanism by implementing and testing additional treatments of this part of the equation of state, including the work of [140] and [154], as well as more recent work by Lattimer & Swesty.

0.10.3 Non-Equilibrium Regions and Thermonuclear Reaction Networks

In CHIMERA, for regions not in NSE, the nuclear composition is evolved using the XNet thermonuclear reaction network code [72]. The numerical methods of XNet will be detailed in § 0.10.3. In CHIMERA v1.0, this fully implicit general purpose reaction network employes only reactions linking the 14 alpha nuclei from ^{4}He to ^{60}Zn. Data for these reactions is drawn from the REACLIB compilation [133]. An additional iron-like nucleus is included in the composition, but not the reaction network, to accommodate a neutron-rich freezeout, which the alpha network is unable to follow, as all included nuclei have equal numbers of neutrons and protons. The abundances of free protons and neutrons are similarly tracked. The advection of material across an NSE - nonNSE interface in either direction, as well as the transition into NSE and freezeout from NSE of entire zones, includes the appropriate gain or loss of nuclear binding energy. The equation of state for the nuclear component in non-NSE regions is a component of the [45] EOS, assuming an ideal gas of nucleons and nuclei whose composition is provided by the nuclear reaction network.

While this 14 element reaction network is sufficient to schematically follow the major nuclear burning stages encountered in the supernova, it is insufficient to study the detailed supernova nucleosynthesis as it is revealed in astronomical observations and terrestrial composition. Neither is the representative nucleus scheme of [45] and [91] sufficient for this purpose. Detailed nucleosynthesis requires evolving 150 or more nuclear isotopes throughout the matter which is ultimately ejected. For reasons we will detail latter in this section, simple replacement of the 14 element network with 150 isotopes in the fully implicit Backward Euler integration scheme represents a several

hundred-fold increase in the computational cost. Given the current 5% of execution time required by the alpha network in the current simulations, this cost is prohibitive. To explore the detailed nucleosynthesis with CHIMERA v1.0 we will adopt a post-processing approach, which is widely used in supernova nucleosynthesis calculations [145, 118]. In this approach, a simplified approximation (in our case an α-network) is used to calculate the compositional change and resulting rate of nuclear energy generation within the hydrodynamic model. Subsequently, the temperature and density histories for individual mass elements from the supernova model are then used as input for separate larger nucleosynthesis calculations. Performing post-processing calculations based on one dimensional models is relatively straightforward, since most one dimensional astrophysical simulations are Lagrangian. Thus the needed temporal histories of temperature and density are simply those of the individual Lagrangian mass elements. However, in multi-dimensions, non-smooth fluid motions result in highly tangled Lagrangian grids. As a result, Eulerian hydrodynamics, where the discretization occurs in space rather than mass, is used in CHIMERA and most multidimensional stellar astrophysics simulations. Because Eulerian codes use spatial discretization, the Lagrangian thermodynamic histories which are a natural result in a Lagrangian code are unavailable. For this reason we are implementing a tracer particle approach, using predictor-corrector integration for the particle path, within our supernova neutrino radiation hydrodynamics models. Integration along the thermodynamic histories gathered from these tracers using XNet and a large reaction network, will allow detailed assessment of the composition of the supernovae ejecta in our models.

Thermonuclear Reaction Rates

Composed of a system of first order differential equations, the nuclear reaction network has sink and source terms representing each of the many nuclear reactions involved. Prior to discussing the numerical difficulties posed by the nuclear reaction network, it is necessary to understand the sets of equations we are attempting to solve. To this end, we present a brief overview of the thermonuclear reaction rates of interest and how these rates are assembled into the differential equations that must ultimately be solved. For more detailed information, we refer the reader to several textbooks covering this subjects [43, 137, 6].

There are a large number of types of nuclear reactions which are of astrophysical interest. In addition to the emission or absorption of nuclei and nucleons, nuclear reactions can involve the emission or absorption of photons (γ-rays) and leptons (electrons, neutrinos, and their anti-particles). As a result, nuclear reactions involve three of the four fundamental forces, the nuclear strong, electromagnetic and nuclear weak forces. Reactions involving leptons (termed weak interactions) proceed much more slowly than those involving only nucleons and photons; however, these reactions are important because only weak interactions can change the global ratio of protons to neutrons.

The most basic piece of information about any nuclear reaction is the nuclear cross section. The cross section for a reaction between target j and projectile k is defined by

$$\sigma = \frac{\text{number of reactions target}^{-1}\text{sec}^{-1}}{\text{flux of incoming projectiles}} = \frac{r/n_j}{n_k v}. \tag{224}$$

The second equality holds when the relative velocity between targets of number density n_j and projectiles of number density n_k is constant and has the value v. Then r, the number of reactions per and sec, can be expressed as $r = \sigma v n_j n_k$. More generally, the targets and projectiles have distributions of velocities, in which case r is given by

$$r_{j,k} = \int \sigma(|\vec{v}_j - \vec{v}_k|)|\vec{v}_j - \vec{v}_k| d^3 n_j d^3 n_k. \tag{225}$$

The evaluation of this integral depends on the types of particles and distributions which are involved. For nuclei j and k in an astrophysical plasma, Maxwell-Boltzmann statistics generally apply; thus,

$$d^3 n = n(\frac{m}{2\pi k_B T})^{3/2} \exp(-\frac{mv^2}{2k_B T}) d^3 v, \tag{226}$$

allowing n_j and n_k to be moved outside of the integral. Eq. 225 can then be written as $r_{j,k} =_{j,k} n_j n_k$, where is the velocity integrated cross section. Equivalently, one can express the reaction rate in terms of a mean lifetime of particle j against destruction by particle k,

$$\tau_k(j) = \frac{1}{{}_{j,k} n_k} \tag{227}$$

For thermonuclear reactions, these integrated cross sections have the form [43, 52]

$$\langle j,k \rangle \equiv_{j,k} = (\frac{8}{\mu\pi})^{1/2} (k_B T)^{-3/2} \int_0^\infty E\sigma(E) \exp(-E/k_B T) dE, \tag{228}$$

where μ denotes the reduced mass of the target-projectile system, E the center of mass energy, T the temperature and k_B is Boltzmann's constant.

Experimental measurements and theoretical predictions for these reaction rates provide the data input necessary to study astrophysical thermonuclear kinetics. While detailed discussion of individual rates is beyond the scope of this report, the interested reader is directed to the following reviews. Experimental nuclear rates and the methodology for measuring them have been reviewed in detail by [137, 83]. The charged particle reaction rates which are important in later stages of stellar evolution and supernovae are reviewed by [28]. Experimental neutron capture cross sections are also summarized by [82, 9]. Theoretical modeling of these rates [47, 97] is vitally important to provide the many rates for which experimental information is incomplete or nonexistent.

When particle k in Eq. 225 is a photon, the distribution $d^3 n_k$ is given by the Plank distribution. Furthermore, the relative velocity is always c and thus the integral is separable, simplifying to

$$r_j = \frac{\int d^3 n_j}{\pi^2 (c\hbar)^3} \int_0^\infty \frac{c\sigma(E_\gamma) E_\gamma^2}{\exp(E_\gamma/k_B T) - 1} dE_\gamma \equiv \lambda_{j,\gamma}(T) n_j. \tag{229}$$

In practice it is not usually necessary to directly evaluate the photodisintegration cross sections [for exceptions see,[112]], because they can be expressed by detailed balance in terms of the capture cross sections for the inverse reaction, $l + m \rightarrow j + \gamma$ [52] .

$$\lambda_{j,\gamma}(T) = (\frac{G_l G_m}{G_j})(\frac{A_l A_m}{A_j})^{3/2}(\frac{m_u k_B T}{2\pi\hbar^2})^{3/2}\langle l,m\rangle \exp(-Q_{lm}/k_B T). \quad (230)$$

This expression depends on the partition functions, $G_k = \sum_i (2J_i{+}1)\exp(-E_i/k_B T)$ (which account for the populations of the excited states of the nucleus), the mass numbers, A, the temperature T, the inverse reaction rate $\langle l,m\rangle$, and the reaction Q-value (the energy released by the reaction), $Q_{lm} = (m_l + m_m - m_j)c^2$. Since photodisintegrations are endoergic, their rates are vanishingly small until sufficient photons exist in the high energy tail of the Planck distribution with energies $> Q_{lm}$. As a rule of thumb this requires $T{\approx}Q_{lm}/30k_B$.

In practice, these experimental and theoretical reaction rates are determined for bare nuclei, while in astrophysical plasmas, these reactions occur among a background of other nuclei and electrons. As a result of this background, the reacting nuclei experience a Coulomb repulsion modified from that of bare nuclei. For high densities and/or low temperatures, the effects of this screening of reactions becomes very important. Under most conditions (with non-vanishing temperatures) the generalized reaction rate integral can be separated into the traditional expression without screening [Eq. 228] and a screening factor,

$$\langle j,k\rangle^* = f_{scr}(Z_j, Z_k, \rho, T, n_i)\langle j,k\rangle. \quad (231)$$

This screening factor is dependent on the charge of the involved particles, the density, temperature, and the composition of the plasma. For more details on the form of f_{scr}, see, , [138, 48, 76, 23].

In stellar plasmas, target nuclei also do not exist solely in their ground states. This complicates the rate expression in Eq. 228, which now must take into account the cross sections for capture out of the different excited states and properly weight them according to their probability of occurrence in the ensemble of target nuclei. Because the timescales for transitions between excited states of a nucleus are typically much shorter than other reaction timescales, it is usually valid to assume that the nuclei are internally equilibrated and a given excited state is populated in the ensemble by the usual Boltzmann factor $\propto e^{-E/k_B T}$, where E now is the excitation energy of that state. From this, one may derive a factor, called the *stellar enhancement factor* (SEF), to correct the ground-state reaction rate for the population of excited states [see, e.g.,[133, 2]].

A procedure similar to that used to derive Eq. 229 applies to captures of electrons by nuclei. Because the electron is 1836 times less massive than a nucleon, the velocity of the nucleus j in the center of mass system is negligible in comparison to the electron velocity ($|\vec{v}_j - \vec{v}_e| \approx |\vec{v}_e|$). In the neutral, completely ionized plasmas typical of the astrophysical sites of nucleosynthesis, the electron number density, n_e, is equal to the total density of protons in nuclei, $\sum_i Z_i n_i$. However in many of these astrophysical settings the electrons are at least partially degenerate, therefore the electron distribution cannot be assumed to be Maxwellian. Instead the capture cross section may be

integrated over a Boltzmann, partially degenerate, or degenerate Fermi distribution of electrons, depending on the astrophysical conditions. The resulting electron capture rates are functions of T and n_e, $r_j = \lambda_{j,e}(T, n_e)n_j$. Similar equations apply for the capture of positrons which are in thermal equilibrium with photons, electrons, and nuclei. Electron and positron capture calculations have been performed for a large variety of nuclei with mass numbers between A=20 and A=100 [for more information see, [99]]. For normal decays, like beta or alpha decays, with a characteristic half-life $\tau_{1/2}$, Eq. 229 also applies, with the decay constant $\lambda_j = \ln 2/\tau_{1/2}$. In addition to innumerable experimental half-life determinations, beta-decay half-lives for a wide range of unstable nuclei have been predicted [22, 113].

Even though the size of the neutrino scattering cross section on nuclei and electrons is very small, in supernovae, the large neutrino flux causes enough events to occur to effect the nuclear composition. Under such conditions the inverse process to electron capture (neutrino capture) occurs in significant numbers and the neutrino capture rate can be expressed in a form similar to Eqs. 229 by integrating over the neutrino distribution [62]. Inelastic neutrino scattering on nuclei can also be expressed in this form [80]. The latter can cause particle emission, similar to photodisintegration. Treatment of these processes in the NSE regions is treated fully self-consistently in the MGFLD code through tabulated rates for these processes folded over a detailed NSE distribution. In the outer, non-NSE regions sufficient captures and scattering occur to affect the matter composition, but the impact of the neutrino luminosity and distribution is negligible. Thus neutrinos reactions should be included in the thermonuclear reaction network, but the corresponding transport opacities may be neglected.

Thermonuclear Rate Equations

The large number of reaction types discussed in §0.10.3 can be divided into 3 functional categories based on the number of reactants which are nuclei. The reactions involving a single nucleus, which include decays, electron and positron captures, photodisintegrations, and neutrino induced reactions, depend on the number density of only the target species. For reaction involving two nuclei, the reaction rate depends on the number densities of both target and projectile nuclei. There are also a few important three-particle process (like the triple-α process) which are commonly successive captures with an intermediate unstable target. Using an equilibrium abundance for the unstable intermediate, the contributions of these reactions are commonly written in the form of a three-particle processes, depending on a trio of number densities. Grouping reactions by these 3 functional categories, the time derivatives of the number densities of each nuclear species in an astrophysical plasma can be written in terms of the reaction rates, r, as

$$\left.\frac{\partial n_i}{\partial t}\right|_{\rho=const} = \sum_j \mathcal{N}_j^i r_j + \sum_{j,k} \mathcal{N}_{j,k}^i r_{j,k} + \sum_{j,k,l} \mathcal{N}_{j,k,l}^i r_{j,k,l}, \qquad (232)$$

where the three sums are over reactions which produce or destroy a nucleus of species i with 1, 2 & 3 reactant nuclei, respectively. The $\mathcal{N}$ s provide for proper accounting of numbers of nuclei and are given by: $\mathcal{N}_j^i = N_i$, $\mathcal{N}_{j,k}^i = N_i/\prod_{m=1}^{n_{j,k}} |N_m|!$, and $\mathcal{N}_{j,k,l}^i = N_i/\prod_{m=1}^{n_{j,k,l}} |N_m|!$. The $N_i's$ can be positive or negative numbers that specify

how many particles of species i are created or destroyed in a reaction, while the denominators, including factorials, run over the $n_{j,k}$ or $n_{j,k,l}$ different species destroyed in the reaction and avoid double counting of the number of reactions when identical particles react with each other (for example in the ^{12}C + ^{12}C or the triple-α reactions; for details see [52]).

In addition to nuclear reactions, expansion or contraction of the plasma can also produce changes in the number densities n_i. To separate the nuclear changes in composition from these hydrodynamic effects, the nuclear abundance $Y_i = n_i/\rho N_A$, where N_A is Avogadro's number, is the quantity most commonly integrated. By recasting Eq. 232 in terms of nuclear abundances Y_i, a set of ordinary differential equations for the evolution of $\dot{Y}_i$ results which depends only on nuclear reactions. In terms of the reaction cross sections introduced in §0.10.3, this reaction network is described by the following set of differential equations

$$\dot{Y}_i = \sum_j \mathcal{N}^i_j \lambda_j Y_j + \sum_{j,k} \mathcal{N}^i_{j,k} \rho N_A \langle j,k \rangle Y_j Y_k + \sum_{j,k,l} \mathcal{N}^i_{j,k,l} \rho^2 N_A^2 \langle j,k,l \rangle Y_j Y_k Y_l. \quad (233)$$

Solving the Nuclear Reaction Network

In principle, the initial value problem presented by the nuclear reaction network for an isolated region can be solved by any of a large number of methods discussed in the literature. However the physical nature of the problem, reflected in the λ's and 's, greatly restricts the optimal choice. The large number of reactions display a wide range of reaction timescales, τ (see Eq. 227). Numerical systems whose solutions depend on a wide range of timescales are termed *stiff*. [65] demonstrated that even a single equation can be stiff if it has both rapidly and slowly varying components. Practically, stiffness occurs when the limitation of the timestep size is due to numerical stability rather than accuracy. A more rigorous definition [89] is that a system of equations $\dot{\vec{Y}}(\vec{Y})$ is stiff if the eigenvalues λ_j of the Jacobian $\partial\dot{\vec{Y}}/\partial\vec{Y}$ obey the criterion that for negative $\Re(\lambda_j)$ (the real part of the eigenvalues λ_j)

$$\mathcal{S} = \frac{max|\Re(\lambda_j)|}{min|\Re(\lambda_j)|} \gg 1 \quad (234)$$

for $j = 1, \cdots, N$. As we will explain in this section, $\mathcal{S} > 10^{15}$ is not uncommon in astrophysics.

Astrophysical calculations of nucleosynthesis belong to the general field of reactive flows, and therefore share some characteristics with related terrestrial fields. In particular, chemical kinetics, the study of the evolution of chemical abundances, is an important part of atmospheric and combustion physics and produces sets of equations much like Eq. 232 [for a through introduction see, [124]]. These chemical kinetics systems are known for their stiffness and a great deal of effort has been expended on developing methods to solve these equations. Many of the considerations for the choice of solution method for chemical kinetics also apply to thermonuclear kinetics. In both cases, temporal integration of the reaction rate equations is broken up into short intervals because of the need to update the hydrodynamics variables. This favors one

step, self starting algorithms. Because abundances must be tracked for a large number of computational cells (hundreds to thousands for one dimensional models, millions to billions for the coming generation of three dimensional models), memory storage concerns favor low order methods since they don't require the storage of as much data from prior steps. In any event, both the errors in fluid dynamics and in the reaction rates are typically a few percent or more, so the greater precision of these higher order methods often does not result in greater accuracy.

Because of the wide range in timescales between strong, electromagnetic and weak reactions, nuclear reaction networks are extraordinarily stiff. PP chain nucleosynthesis, responsible for the energy output of the Sun, offers an excellent example of the difficulties. The first reaction of the PP1 chain is ${}^1H(p,e^+\nu){}^2H$, the fusion of two protons to form deuterium. This is a weak reaction, requiring the conversion of a proton into a neutron, and releasing a positron and a neutrino. As a result, the reaction timescale $\tau_p({}^1H)$ is very long, billions of years for conditions like those in the solar interior. The second reaction of the PP1 chain is the capture of a proton on the newly formed deuteron, ${}^2H(p,\gamma){}^3He$. For conditions like those in the solar interior, the characteristic timescale, $\tau_p({}^2H)$ is a few seconds. Thus the timescales for two of the most important reactions for hydrogen burning in stars like our Sun differ by more than 17 orders of magnitude [for a more complete discussion of the PP chain see, [43, 68]]. This disparity results not from a lack of ^{1}H + ^{1}H collisions (which occur at a rate $Y({}^1H)/Y({}^2H) \sim 10^{17}$ times more often than ^{1}H + ^{2}H collisions), but from the rarity of the transformation of a proton to a neutron. While the presence of weak reactions among the dominant energy producing reactions is unique to hydrogen burning, most nucleosynthesis calculations are similarly stiff, in part because of the need to include weak interactions but also the potential for neutron capture reactions, which occur very rapidly even at low temperature, following any release of free neutrons. The nature of the nuclear reaction network equations has thus far limited the astrophysical usefulness of the most sophisticated methods to solve stiff equations developed for chemical kinetics. For example, [114] have shown that asymptotic and quasi-steady state methods have very limited success integrating astrophysical reaction networks. However work to harness this resource continues.

For a set of nuclear abundances $\vec{Y}$, one can calculate the time derivatives of the abundances, $\dot{\vec{Y}}$ using Eq. 233. The desired solution is the abundance at a future time, $\vec{Y}(t+\Delta t)$, where Δt is the network timestep. For simplicity, most past and present nucleosynthesis calculations use the simple finite difference prescription

$$\frac{\vec{Y}(t+\Delta t)-\vec{Y}(t)}{\Delta t} = (1-\Theta)\dot{\vec{Y}}(t+\Delta t) + \Theta\dot{\vec{Y}}(t). \tag{235}$$

This choice is also supported by the advantages of coupling low order, single step methods with hydrodynamics. With $\Theta = 1$, Eq. 235 becomes the explicit Euler method while for $\Theta = 0$ it is the implicit backward Euler method, both of which are first order accurate. For $\Theta = 1/2$, Eq. 235 is the semi-implicit trapezoidal method, which is second order accurate. XNet uses the fully implicit, backward Euler as this has been most reliable for solving the stiff set of non-linear differential equations which form most nuclear reaction networks, particularly those approaching equilibrium. Solving

the fully implicit version of Eq. 235 is equivalent to finding the zeros of the set of equations

$$\vec{\mathcal{Z}}(t+\Delta t) \equiv \frac{\vec{Y}(t+\Delta t) - \vec{Y}(t)}{\Delta t} - \dot{\vec{Y}}(t+\Delta t) = 0 \; . \tag{236}$$

This is done using the Newton-Raphson method [127], which is based on the Taylor series expansion of $\vec{\mathcal{Z}}(t+\Delta t)$, with the trial change in abundances given by

$$\Delta\vec{Y} = \left(\frac{\partial \vec{\mathcal{Z}}(t+\Delta t)}{\partial \vec{Y}(t+\Delta t)} \right)^{-1} \vec{\mathcal{Z}} \; , \tag{237}$$

where $\partial\vec{\mathcal{Z}}/\partial\vec{Y}$ is the Jacobian of $\vec{\mathcal{Z}}$.

Historically [5] and in some modern applications [84], each timestep consists of only a single application of the procedure outlined in Eqs. 236 & 237. This semi-implicit backward Euler method has the advantage of a relatively small and predictable number of matrix solutions, but there are only heuristic checks that the chosen timestep results in a stable or accurate solution. For fully implicit backward Euler schemes like XNet, iteration using the procedure of Eqs. 236 & 237 continues until both $\Delta\vec{Y}$ and $\vec{\mathcal{Z}}$ are below some tolerance, providing a measure of the stability and accuracy. If this convergence does not occur within a reasonable number of iterations, the timestep is subdivided into smaller intervals until a converged solution can be achieved, allowing the fully implicit backward Euler integration to respond to instability or inaccuracy in a way that is impossible with the semi-implicit backward Euler approach. Higher order methods allow better estimates of the truncation error by comparing the solutions of different order schemes and sub-dividing the timestep if these errors are too large.

A potential numerical problem with the solution of Eq. 236 is the singularity of the Jacobian matrix, $\partial\vec{\mathcal{Z}}(t+\Delta t)/\partial\vec{Y}(t+\Delta t)$. From Eq. 236, the individual matrix elements of the Jacobian have the form

$$\frac{\partial \mathcal{Z}_i}{\partial Y_j} = \frac{\delta_{ij}}{\Delta t} - \frac{\partial \dot{Y}_i}{\partial Y_j} = \frac{\delta_{ij}}{\Delta t} - \sum \frac{1}{\tau_j(i)} \; ,$$

where δ_{ij} is the Kronecker delta, and $\tau_j(i)$ is the destruction timescale of nucleus i with respect to nucleus j for a given reaction, as defined in Eq. 227. The sum accounts for the fact that there may be more than one reaction by which nucleus j is involved in the creation or destruction of nucleus i. Along the diagonal of the Jacobian, there are two competing terms, $1/\Delta t$ and $\sum 1/\tau_i(i)$. This sum is over all reactions which destroy nucleus i, and is dominated by the fastest reactions. As a result, $\sum 1/\tau_i(i)$ can be orders of magnitude larger than the reciprocal of the desired timestep, $1/\Delta t$. This is especially a problem near equilibrium, where both destruction and the balancing production timescales are very short in comparison to the preferred timestep size, resulting in differences close to the numerical accuracy (i.e. 14 or more orders of magnitude). In such cases, the term $1/\Delta t$ is numerically neglected, leading to numerically singular matrices. One historical approach to avoiding this problem is to artificially scale these short, equilibrium timescales by a factor which brings their timescale closer to Δt, but leaves them small enough to ensure equilibrium. While this approach has been used successfully, the ad hoc nature of this artificial scaling renders these methods

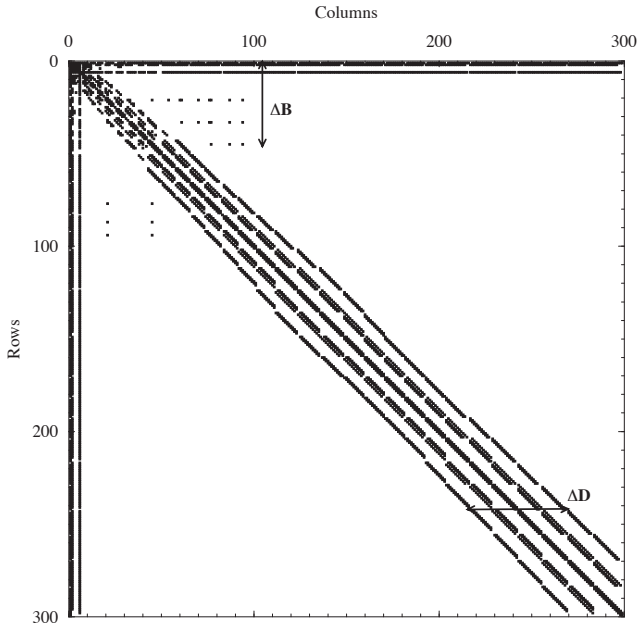


Figure 3: Graphic demonstration of the sparseness of the Jacobian matrix. The filled squares represent the non-zero elements.

fragile. A more promising approach is to make directly use of equilibrium expressions for abundances, which, as we will discuss in §0.10.3, also ensures the economical use of computer resources.

Taking Advantage of Matrix Sparseness

For larger nuclear reaction networks, the Newton-Raphson method requires solution of a moderately large ($N = 100 - 3000$) matrix equation for each zone. Since general solution of a dense matrix scales as $O(N^3)$, this can make these large networks progressively much more expensive. While in principal, every species reacts with each of the hundreds of others, resulting in a dense Jacobian matrix, in practice it is possible to neglect most of these reactions. Because of the $Z_i Z_j$ dependence of the repulsive Coulomb term in the nuclear potential, captures of free neutrons and isotopes of H and He on heavy nuclei occur much faster than fusions of heavier nuclei. Furthermore, with the exception of the Big Bang nucleosynthesis and PP-chains, reactions involving secondary isotopes of H (deuterium and tritium) and He are neglectable. Likewise, photodisintegrations tend to eject free nucleons or α-particles. Thus, with a few important exceptions, for each nucleus we need only consider twelve reactions linking it

to its nuclear neighbors by the capture of an n, p, α or γ and release a different one of these four. The exceptions to this rule are the few heavy ion reactions important for burning stages like carbon and oxygen burning where the dearth of light nuclei cause the heavy ion collisions to dominate.

Fig. 3 demonstrates the sparseness of the resulting Jacobian matrix, for a 300 species network chosen to handle all the energy generating stages in the life of a massive star. The nuclei are indexed in order of increasing Z and then A. Of the 90,000 matrix elements, fewer than 5,000 are non-zero. In terms of the standard forms for sparse matrices, this Jacobian is best described as doubly bordered, band diagonal. With a border width, ΔB, of 45 necessary to include the heavy ion reactions among ^{12}C, ^{16}O and ^{20}Ne along with the free neutrons, protons and α -particles and a band diagonal width, ΔD, of 54, even this sparse form includes almost 50,000 elements. With solution of the matrix equation consuming 90+% of the computational time, there is clearly a need for custom tailored solvers which take better advantage of the sparseness of the Jacobian [126]. To date best results for small to medium sized (N$<$200) matrices are obtained with machine optimized dense solvers (LAPACK) or matrix specific solvers generated by symbolic processing [116, 115]. For larger matrices, generalized sparse solvers, both custom built and from software libraries, are used [146]. XNet includes the option of using the MA28 or PARDISO sparse matrix packages for large reaction networks, but they do not provide significant advantages for matrices of the sizes used in CHIMERA 1.0 or anticipated for CHIMERA 2.0.

Coupling Nuclear Reaction Networks to Hydrodynamics

Nuclear processes are tightly linked to the hydrodynamic behavior of the bulk medium. Thermonuclear processes release (or absorb) energy, which alters the pressure and causes hydrodynamic motions. These motions may disperse the thermonuclear ash and bring a continued supply of fuel to support the thermonuclear flame. The compositional changes caused by thermonuclear reactions can also change the equation of state and opacity, further impacting the hydrodynamic behavior. In CHIMERA, as in all current multi-dimensional astrophysical simulations, these thermonuclear and hydrodynamic changes in local composition are treated in an operator split fashion. [116] provides an authoritative overview and discusses the difficulties (and open issues) involved when including nucleosynthesis within hydrodynamic simulations, but we here summarize the basic issues as they relate to CHIMERA.

The coupling between thermonuclear processes and hydrodynamic changes can be divided into two categories by considering the spatial extent of the coupling. Nucleosynthetic changes in composition and the resultant energy release produce *local* changes in hydrodynamic quantities like pressure and temperature. The strongest of these local couplings is the release (or absorption) of energy and the resultant change in temperature. Changes in temperature are particularly important because of the exponential nature of the temperature dependence of thermonuclear reaction rates. Since the nuclear energy release is uniquely determined by the abundance changes, the rate of thermonuclear energy release, $\dot{\epsilon}$, is given by

$$\dot{\epsilon}_{nuc} = -\sum_i N_A M_i c^2 \dot{Y}_i(). \tag{238}$$

where $M_i c^2$ is the rest mass energy of species i in . Since all reactions conserve nucleon number, the atomic mass excess $M_{ex,i} = M_i - A_i m_u$ (m_u is the atomic mass unit) can be used in place of the mass M_i in Eq. 238 [for a recent compilation of mass excesses see,[7]]. The use of atomic mass units has the added benefit that electron conservation is correctly accounted for in the case of β^- decays and e^- captures, though reactions involving positrons require careful treatment. In general, the nuclear energy release is deposited locally, so the rate of thermonuclear energy release is equal to the nuclear portion of the hydrodynamic heating rate. However, there are instances where nuclear products do not deposit their energy locally. Escaping neutrinos can carry away a portion of the thermonuclear energy release. In the rarefied environment of supernova ejecta at late times, positrons and gamma rays released by β decays are not completely trapped. In most such cases, the escaping particles stream freely from the reaction site, allowing adoption of a simple loss term analogous to Eq. 238 with $M_i c^2$ replaced by an averaged energy loss term. For this reason, weak reaction rate tabulations provide averaged neutrino losses. From these we can construct

$$\dot{\epsilon}_{\nu\ loss} = \sum_i \langle E_\nu \rangle \dot{Y}_{i,weak}, \tag{239}$$

where we consider only those contributions to $\dot{Y}$ due to neutrino producing reactions. For the deeper layers of the supernovae, subsequent interactions between the escaping leptons or gamma rays and matter require complete transport to be considered, but for the most part, in the ejecta the emitted neutrinos escape without further interactions. Other important quantities which are impacted by nucleosynthesis, like , can be obtained by appropriate sums over the abundances and also need not be evolved separately.

Implicit solution methods require the calculation of $\dot{Y}(t + \Delta t)$, where Δt is the nuclear timestep, which in turn requires knowledge of $T(t + \Delta t)$. One could write a differential equation for the energy release analogous to Eq. 233, with the $\mathcal{N}$s replaced by the reaction Q-values, and thereby evolve the energy release (and calculate temperature changes) as an additional equation within the network solution. [115] has shown that such a scheme can help avoid instabilities in the case of a physically isolated zone entering or leaving nuclear statistical equilibrium. In general, however, use of this additional equation is made unnecessary by the relative slowness with which the temperature changes. The timescale on which the temperature changes is given by

$$\tau_T = T/\dot{T} \approx C_V T/\dot{\epsilon}_{nuc} \tag{240}$$

and is often called the *ignition timescale*. The timescale on which an individual abundance changes is its *burning time*,

$$\tau({}^A Z) = Y({}^A Z)/\dot{Y}({}^A Z) \sim \min_k \tau_k({}^A Z) \tag{241}$$

where $\tau_k({}^A Z)$ is defined in Eq. 227. In general τ_T differs from $\tau({}^A Z)$ of the principle fuel by the ratio of thermal energy content to the energy released by the reaction. For degenerate matter this ratio can approach zero, allowing for explosive burning. In contrast, accurate prediction of less abundant, but still important, species requires that

the reaction network timestep Δt be chosen to be the burning timescale of a less abundant species, typically with an abundance of 10^{-6} or smaller [5]. Since the dominant fuel is most often one of the more abundant constituents and the burning timescales are proportional to the abundance, τ_T is typically an order of magnitude or more larger than the reaction network timestep [150, 13]. It is therefore sufficient to calculate the energy gain at the end of a timestep via equation 238, modified as discussed above, and approximate $T(t + \delta t) \approx T(t)$ or to extrapolate based on (t). Since other locally coupled quantities have characteristic timescales much longer than Δt, they too can be decoupled in a similar fashion.

Spatial coupling, particularly the modification of the composition by hydrodynamic movements such as diffusion, convective mixing and advection (in the case of Eulerian hydrodynamics methods), represents a more difficult challenge. By necessity, an individual nucleosynthesis calculation examines the abundance changes in a locality of uniform composition. The difficulties associated with strong spatial coupling of the composition occur because this nucleosynthetic calculation is spread over an entire hydrodynamic zone. Convection and other fluid movements can result in strong abundance gradients across a single hydrodynamic zone, which with the assumption of compositional uniformity, can result in very different outcomes as a function of the fineness of the hydrodynamic grid. This is a much bigger issue for thermonuclear supernovae, where the thickness of the flame front can be billionths of the radius of the white dwarf [121, 85]. Eulerian advection of compositional boundaries can also have extremely unphysical consequences. [61] demonstrated how this artificial mixing can produce an unphysical detonation in a shock tube calculation by mixing cold unburnt fuel into the hot burnt region. A related problem is the conservation of species. Hydrodynamic schemes must carefully conserve the abundances (or partial densities) of all species [61, 90, 125], lest they provide unphysical abundances to the nucleosynthesis calculations, which must assume conservation, and thereby produce unphysical results. Because of these problems, hydrodynamic methods with excellent capture of shocks and contact or compositional discontinuities are best suited to nucleosynthesis calculations. This is, in part, the motivation for the use of PPM hydrodynamics in CHIMERA. To maximize this accuracy, the Consistent Multifluid Advection (CMA) scheme of [125] has been implemented in CHIMERA.

The relative size of the burning timescales, when compared to the relevant diffusion, sound crossing or convective timescale, dictates how these problems must be addressed. If all of the burning timescales are much shorter than the timescale on which the hydrodynamics changes the composition, then the assumption of uniform composition is satisfied and the nucleosynthesis of each hydrodynamic zone can be treated independently. If all of the burning timescales are much larger than, for example, the convective timescale, then the composition of the entire convective region can be treated as uniform and slowly evolving. The greatest complexity occurs when the timescales on which the hydrodynamics and nucleosynthesis change the composition are similar. Hydrostatic oxygen shell burning represents an excellent example of this as the sound travel, convective turnover and nuclear burning timescales are all of the same order as the evolutionary time. The results of 2D simulations [12, 101] demonstrate convective overshooting, highly non-uniform burning and a velocity structure dominated by convective plumes. Hydrostatic silicon burning represents a different challenge [6], as the

timescales for the transformation of silicon to iron are much slower than the convective turnover time, but the burning timescales for the free neutrons, protons and α-particles which maintain QSE are much faster, providing a strong motivation for the hybrid reaction networks we will discuss in §0.10.3. In the cases of explosive oxygen and silicon burning in core collapse supernovae, the driving of the nuclear burning by the shock wave results in much faster nuclear burning timescales weakening the coupling between burning and mixing.

Toward *in-situ* nucleosynthesis: the QSE-reduced Network

As discussed previously, the use of a 14 element α-network within CHIMERA v1 models is a compromise between execution time and physical accuracy for the detailed nucleosynthesis. Even with the larger post-processing calculations using tracer particles providing abundance information for a wider variety of species, questions will remain because post-processing, though widely used in supernova nucleosynthesis calculations, has significant drawbacks. The primary limitation in a post-processing approach is the accuracy of the energy generation rate provided by the approximation included within the hydrodynamics. For energetically unimportant supernova nucleosynthesis processes, like the ν, p- and r-processes, post-processing is probably adequate. Indeed, given uncertainties in how macroscopic mixing on the scale of the Eulerian grid translates into microscopic mixing at the nuclear reaction scale, post-processing is potentially beneficial for these processes. For oxygen, neon and carbon burning in the outer layers of core, the nuclear energy release is a fraction (10-20%) of the energy imparted by the shock, however our recent work [27, 105] suggests the coupling of this nuclear energy release to the hydrodynamic instabilities can provide a much larger impact. For the innermost supernova ejecta, the nuclear energy released by the recombination of α-particles into iron (and neighboring) nuclei $(1 - 2 \times 10^{18})$ is comparable to the change in the thermal energy of this gas due to expansion during the same time, thus there is significant feedback between the rate of this nuclear recombination and the temperature evolution which is driving the recombination. The α-richness of the matter, and thus the abundance of species like ^{44}Ti, ^{57}Fe, ^{58}Ni and ^{60}Zn [157], therefore depends critically on this feedback. These qualitative arguments for the need to better couple modeling of the supernova mechanism and the resulting nucleosynthesis are backed up by recent simulations [86, 129, 128, 55, 56] which show that consideration of the effects of neutrino interactions, convection and other important features of the mechanism alters the production and distribution of heavy elements in the ejecta. In spite of the computational cost, only by extending CHIMERA, and competing models, to include nucleosynthesis beyond an α-network can these impacts be fully understood.

Recently developed methods should accelerate our ability to do this. [71] have achieved factors of 5-10 decrease in the computational cost of the network during the most computationally expensive supernova burning phases by employing local partial equilibria (termed quasi-equilibrium or QSE) to reduce size of the system of equations which must be integrated. With modest further refinement, these QSE-reduced networks will enable self-consistent nucleosynthesis within the radiation-transport problem at a cost only a few times the current simulations. As we noted in the previous section, while global NSE may not always apply for temperatures in the range 3-6 ,

many nuclei are in local equilibrium with their neighbors. Beginning with [21], a number of attempts have been made to take advantage of these partial equilibria (termed quasi-equilibria or QSE) to reduce the number of independent variables evolved via rate equations and thereby reduce the computational cost of modeling these burning stages. To evolve the abundances of every member of a QSE group, it is sufficient to evolve the abundance of any single group member along with the abundances of the free nucleons. One can thereby reduce the number of abundances that are evolved, while still calculating, from QSE relations, the abundances of all members of a QSE group and the resultant reaction rates, including the electron and neutrino capture reactions responsible for changes in the neutronization. The result is a more computationally efficient method that retains the accuracy of the full network and yields abundances for all nuclei found in the full network. Such methods have been applied to the α-network and produced networks that are twice as fast as the minimal α-chain network, without significantly affecting the nuclear evolution [70, 147]. Reductions of an order of magnitude in computational cost have been achieved [54, 71] for QSE-reduced networks of the size necessary to capture the essential features of supernova nucleosynthesis and we believe greater savings are possible with further refinement.

As an example, we will discuss the QSE-reduced α-network. Its mission is to evolve the abundances of the full 14 elements of a conventional α-network (which we'll call $\vec{Y}^{\mathcal{F}}$), and calculate the resulting energy generation, in a more efficient way. Under conditions where QSE applies, the existence of the silicon and iron peak QSE groups (which are separated by the nuclear shell closures Z=N=20 and the resulting small Q-values and reaction rates) allows calculation of these 14 abundances from 7. For the members of the silicon group (^{28}Si, ^{32}S, ^{36}Ar, ^{40}Ca, ^{44}Ti) and the iron peak group (^{48}Cr, ^{52}Fe, ^{56}Ni, ^{60}Zn) the individual abundances can be calculated by expressions similar to Eq. 223,

$$\begin{aligned} Y_{QSE,\mathrm{Si}}(^{A}Z) &= \frac{C(^{A}Z)}{C(^{28}Si)} Y(^{28}Si) Y_{\alpha}^{\frac{A-28}{4}} \\ Y_{QSE,\mathrm{Ni}}(^{A}Z) &= \frac{C(^{A}Z)}{C(^{56}Ni)} Y(^{56}Ni) Y_{\alpha}^{\frac{A-56}{4}}, \end{aligned} \tag{242}$$

where $C(^{A}Z)$ is defined in Eq. 223 and $(A-28)/4$ and $(A-56)/4$ are the number of α-particles needed to construct ^{A}Z from ^{28}Si and ^{56}Ni, respectively. Where QSE applies, $\vec{Y}^{\mathcal{F}}$ is a function of the abundances of a reduced nuclear set, $\mathcal{R}$, defined as α, ^{12}C, ^{16}O, ^{20}Ne, ^{24}Mg, ^{28}Si, ^{56}Ni and we need only evolve $\vec{Y}^{\mathcal{R}}$. It should be noted that ^{24}Mg is ordinarily a member of the silicon QSE group [6, 156, 72], but for easier integration of prior burning stages with a conventional nuclear network, we will evolve ^{24}Mg independently. The main task when applying such hybrid schemes is finding the boundaries of QSE groups and where individual nuclei have to be used instead. Treating marginal group members as part of a group increases the efficiency of the calculation, but may decease the accuracy.

While $\vec{Y}^{\mathcal{R}}$ is a convenient set of abundances for calculating $\vec{Y}^{\mathcal{F}}$, it is not the most efficient set to evolve, primarily because of the non-linear dependence on Y_{α}. Instead we define a set of group abundances, $\vec{Y}^{\mathcal{G}}$= [$Y_{\alpha G}$, $Y(^{12}C)$, $Y(^{16}O)$, $Y(^{20}Ne)$,

$Y(^{24}Mg)$, Y_{SiG}, Y_{FeG}] where

$$\begin{aligned} Y_{\alpha G} &= Y_\alpha + \sum_{i \in Si\ group} \frac{A_i - 28}{4} Y_i + \sum_{i \in Fe\ group} \frac{A_i - 56}{4} Y_i \ , \\ Y_{SiG} &= \sum_{i \in Si\ group} Y_i \ , \\ Y_{FeG} &= \sum_{i \in Fe\ group} Y_i \ . \end{aligned} \tag{243}$$

Physically, $Y_{\alpha G}$ represents the sum of the abundances of free α-particles and those α-particles required to build the members of the QSE groups from ^{28}Si or ^{56}Ni, while Y_{SiG} and Y_{FeG} represent the total abundances of the silicon and iron peak QSE groups. This method, which here is applied only to the chain of α-nuclei can also be generalized to arbitrary networks [71]. For larger networks which contain nuclei with $N \neq Z$, one must be able to follow the abundances of free neutrons and protons, particularly since weak interactions will change the global ratio of neutrons to protons. In place of $Y_{\alpha G}$ in Eq. 243, one constructs $Y_{NG} = \sum_{i,light} N_i Y_i + \sum_{i,Si}(N_i - 14)Y_i + \sum_{i,Fe}(N_i - 28)Y_i$ and $Y_{ZG} = \sum_{i,light} Z_i Y_i + \sum_{i,Si}(Z_i - 14)Y_i + \sum_{i,Fe}(Z_i - 28)Y_i$, if ^{28}Si and ^{56}Ni are chosen as the focal nuclei for the Si and Fe groups.

Corresponding to this reduced set of abundances $\mathcal{G}$ is a reduced set of reactions, with quasi-equilibrium allowing one to ignore the reactions among the members of the QSE groups. Unfortunately, the rates of these remaining reactions are functions of the full abundance set, $\vec{Y}^{\mathcal{F}}$, and are not easily expressed in terms of the group abundances, $\vec{Y}^{\mathcal{G}}$. Thus, for each $\vec{Y}^{\mathcal{G}}$, one must solve for $\vec{Y}^{\mathcal{R}}$ and, by Eq. 242, $\vec{Y}^{\mathcal{F}}$, in order to calculate $\dot{\vec{Y}}^{\mathcal{G}}$ which is needed to evolve $\vec{Y}^{\mathcal{G}}$ via Eq. 235. Furthermore, Eq. 237 requires the calculation of the Jacobian of $\vec{\mathcal{Z}}$, which can not be calculated directly since $\dot{\vec{Y}}^{\mathcal{G}}$ cannot be expressed in terms of $\vec{Y}^{\mathcal{G}}$. Instead it is sufficient to use the chain rule,

$$\frac{\partial \dot{\vec{Y}}^{\mathcal{G}}}{\partial \vec{Y}^{\mathcal{G}}} = \frac{\partial \dot{\vec{Y}}^{\mathcal{G}}}{\partial \vec{Y}^{\mathcal{R}}} \frac{\partial \vec{Y}^{\mathcal{R}}}{\partial \vec{Y}^{\mathcal{G}}} \tag{244}$$

to calculate the Jacobian. Analytically, the first term of the chain rule product is easily calculated from the sums of reaction terms, while the second term requires implicit differentiation using Eq. 243. Additional details and comparisons with the full α-network are demonstrated by [70] [see also, [147]]. A generalized implementation is detailed in [71].

0.11 CHIMERA Problem Statement and Results

0.11.1 Q2

We simulate the collapse, bounce, and post-bounce evolution of a 11 solar mass star (progenitor model s11.2 (Woosley, private communication) using CHIMERA. The simulation resolution is 256 radial zones and 256 angular zones in space and uses 20 energy

groups to resolve the neutrino spectra. Nuclear burning is approximated with an alpha network spanning ^{4}He to $^{6}0$Zn using 14 nuclei. The gravitational field in the star is assumed Newtonian.

We performed a benchmark run to test the performance of the code. The benchmark consists of a finite number of evolution timesteps. A full simulation of collapse, bounce, and post-bounce evolution is prohibitive for this purpose, as it requires more than 300 wallclock hours, even given dedicated access. The benchmark consisted of 10 full transport timesteps of the full 256x256 model problem (incorporating all the physics described above). Hydrodynamic subcycling was implemented, with 20 hydrodynamic timesteps per transport step.

The benchmark was executed on jaguar at the NCCS (http://info.nccs.gov/resources/jaguar). 256 "XT3" processors were specified in the job submission.

We consider the wallclock time per hydrodynamic step to be the most reasonable timing metric. Our benchmark run produced 3.06 s/step for the 2D problem.

0.11.2 Q4

We have extended and partially re-engineered CHIMERA to allow for three spatial dimensions to be evolved by the code. We repeated our benchmark simulation from Q2, save only for the fact that we now perform the problem in 3D. All the attendant input physics remain the same as in Q2, as does the input deck. We again performed 10 full transport timesteps of the full model problem (now using 256 radial zones and an angular resolution of 32x64). Hydrodynamic subcycling was implemented, with 20 hydrodynamic timesteps per transport step. 2048 "XT3" processors of jaguar were used.

Epoch	Dimensionality	processes	wallclock time/step (s)
Q2	2	256	3.06
Q4	3	2048	3.25

The new 3D version of CHIMERA is able to evolve a much more realistic (in that it relaxes the assumption of axisymmetry) model of core collapse and explosion than before at essentially the same cost in wallclock time. Our new domain decomposition for the hydrodynamics is the primary reason for this excellent scaling behavior. The new "pencil" decomposition involves 2 smaller transposes (over each directionally-split dimension normal to the current sweep) rather than one global transpose per fractional timestep. These smaller transposes are much more easily handled by the XT network.

0.12 CHIMERA Verification

The verification of the CHIMERA code has proceeded along two major lines: (1) Verification of CHIMERA's individual modules (e.g., the hydrodynamics module). (2) Verification of the CHIMERA simulation output against full, multi-physics, spherically symmetric core collapse supernova simulations performed with the Agile-BOLTZTRAN code [94]. We report on the latter here.

In figure 4, the agreement is evident between CHIMERA and Agile-BOLTZTRAN for the stellar core electron fraction, entropy, and velocity profiles at 200 milliseconds after bounce, which is during the critical shock reheating/revival phase. All three quantities are sensitive measures of the numerical treatment of stellar core neutrino transport and thermodynamics/hydrodynamics. The differences at the shock are largely associated with the different numerical hydrodynamics methods used in the two codes, although some of the differences there, and overall, are associated with the different number of radial zones used in the runs, with CHIMERA deploying 256 zones and Agile-BOLTZTRAN deploying 96 and 192 zones, respectively. CHIMERA is based on the piecewise parabolic method (PPM) and uses a fixed Eulerian grid. PPM, which is based on a Riemann solution at shock interfaces, is ideally suited to flows involving shocks. Agile-BOLTZTRAN is based on a second-order finite difference representation but uses an adaptive radial grid. Generally, the PPM treatment will better resolve shocks and more accurately compute the jump in quantities such as entropy across it. The different electron fraction profiles below 10 km, with CHIMERA generating slightly higher values, is associated with the slightly lower values computed by CHIMERA for the stellar core entropy in this region. The lower core entropy is again a result of the different numerical hydrodynamics methods. A lower entropy leads to a lower proton fraction in this region, which in turn leads to less electron capture and, consequently, higher electron fractions. The different entropy profiles ahead of the shock are the result of different treatments of non-NSE material, with CHIMERA implementing a nuclear network and Agile-BOLTZTRAN treating all non-NSE material as silicon. These minor differences do not have ramifications for the overall core collapse supernova dynamics, which is very clearly the same for both codes.

0.13 Toward the Petascale and Higher-Fidelity 3D Simulations

0.13.1 Architectural Perspective

From a high-level architectural perspective, petascale machines will have multiple nodes, each with a shared memory footprint. Moreover, every processor used in these systems will be a multi-core socket. The interconnect network will have robust bandwidth and a hierarchical topology; some systems may even have multiple networks for specific functionality. This means there will be at least three levels of concurrency to be managed by the software used to construct an application. Most codes today are managed at one level using either MPI or OpenMP. If the petascale system has accelerator technology (FPGA or Cell-based systems) there will be even more complex software interactions required. A schematic petascale system will have 4 to 100 cores on a socket, the node will likely have a shared memory region of 4 to 512 processors, and there will be 30,000 to 100,000 nodes. This will lead to millions of threads of execution to be managed by application software. Our development plan is designed to maximize the efficiency with which CHIMERA can utilize such an architecture within the code architecture we have already established. Our path forward presents a special opportunity for early coding and testing on petascale hardware because CHIMERA

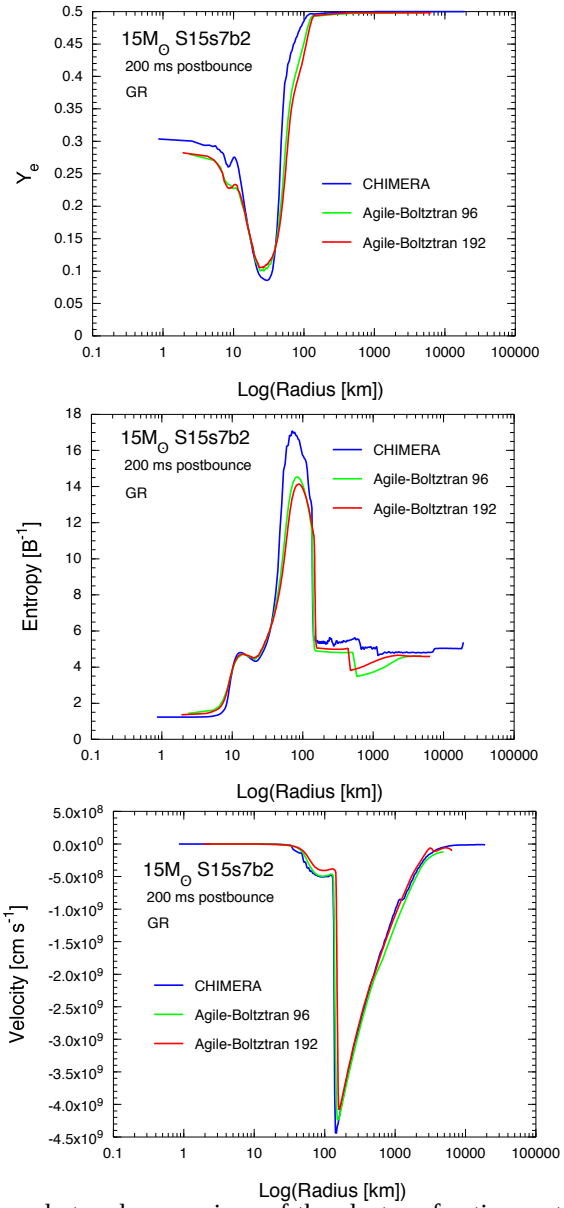


Figure 4: A snapshot and comparison of the electron fraction, entropy, and velocity profiles at 200 ms after stellar core bounce, computed with the CHIMERA and Agile-BOLTZTRAN codes.

will encapsulate a remarkable range of algorithms and a variety of implementations for this type of machine (PGAS languages, MPI, threads) in an integrated whole. This testbed is already well-developed, with a clear path forward, as delineated below.

0.13.2 Science Overview

The existing CHIMERA code will be used to perform the first three-dimensional, multi-physics core collapse supernova simulations with multi-frequency neutrino transport. Low-resolution simulations are currently running on the Cray XT4 at the ORNL Leadership Computing Facility. Higher resolution runs are planned for the 250 TF T2O period in early calendar 2008. The jump to both three spatial dimensions *and* multi-frequency (as opposed to "gray") transport will mark a quantum leap in realism in core collapse supernova modeling. These simulations will be performed with (1) the θ axis excised from the simulation domain to avoid the coordinate singularity there, (2) spherical (as opposed to three-dimensional) gravity, (3) a spherical general relativistic correction to the spherical Newtonian gravitational field, and (4) an "α" network of nuclei consisting of 17 isotopes, which is sufficient to capture the gross energetics of nuclear burning in core collapse supernovae but not the detailed element synthesis (which requires at least 150 isotopes). Following these first three-dimensional simulations, each of the above approximations will be removed and higher-fidelity 3D simulations will be performed. The coordinate singularity at $\theta = 0$ will be removed with the deployment of the Yin–Yang grid; the approximation of Newtonian spherical gravity will be relaxed with a full solution of the three-dimensional Poisson equation and, subsequently, the Poisson solution and spherical correction for general relativity will be replaced by the conformally flat approximation (CFA) and, ultimately, a complete (BSSN) Einstein equation solve; and a 150-isotope network will eventually replace the α network, which will require the development of "quasi-equilibrium (QSE)" techniques to accommodate this many isotopes *in situ* in a three-dimensional model. Also planned is an eventual swap of the transport module in CHIMERA, from multi-frequency flux-limited diffusion (mCHIMERA) to multi-frequency *and* multi-angle Boltzmann transport (bCHIMERA).

0.13.3 Effective Utilization of Multi-Core Sockets

At present, CHIMERA assigns a single MPI task to each core when running on the Cray XT4. As the complexity of the thermonuclear burning and the phase space resolution of the neutrino transport increase, the most efficient way to exploit multiple cores on a socket will be to expose the parallelism inherent in those modules. At that point, each multi-core socket will be assigned to a single MPI task, and the multiple cores will be marshaled to perform the "local" computation inherent in the thermonuclear kinetics and neutrino transport.

In each case, we have two options for implementation, assuming the requisite operating system and language support are available on the petascale platforms: using MPI sub-communicators or using a hybrid MPI/threaded strategy. Each method introduces some degree of overhead, but the amount and impact of that overhead is strongly implementation dependent. Therefore, we intend to investigate both possibilities on

particular hardware platforms. The exploration of these strategies, as in the case of our I/O developments, is concerned with risk mitigation. No matter which is the more efficient implementation, the amount of inherent parallelism still present in our problem will provide ample opportunity for performance enhancement. If, on the other hand, our approach merely rearranged already exposed parallelism (by hybridizing the spatial domain decomposition), our strategy would likely have less chance of success.

The thermonuclear burning module of CHIMERA has a straightforward multi-core implementation. Because each zone in a ray is evolved separately, these tasks can be performed concurrently on individual cores. Furthermore, by performing a cyclic decomposition along a ray (i.e. core 1 might evolve zones 1, 17, 33, etc., while core 2 evolves zone 2, 18, 34, etc.), a natural load balancing can be achieved with no additional work. As the temperature of a zone predominantly determines the amount of computation necessary in the thermonuclear burning module and the temperature profile remains roughly spherically symmetric even in multidimensional simulations, each core will perform approximately the same number of FLOPs using this method.

The neutrino transport module also presents unrealized parallelism in the form of physics-based preconditioning. Many of the non-zeroes in the sparse linear system that stems from linearizing the MGFLD equations occur in diagonal blocks. These blocks represent local weak interaction physics, neutrino scattering on matter components. An effective preconditioner inverts these local blocks (or some approximation to these blocks) (see [46]). As these inversions are independent of one another, they can also be carried out concurrently. We have previously implemented a hybrid MPI/OpenMP preconditioner of this type for MGBT transport [46]. We expect that adapting a version of this preconditioner for the MGFLD module in CHIMERA will be straightforward. The conversion of the preconditoner to use a MPI sub-communicator should also prove relatively straightforward, as the threaded portions of the present code consist of a series of PARALLEL DO sections. More work will center on determining useful and effective sparse approximations to a full LU decomposition of the diagonal blocks.

The replacement of the MGFLD transport module with one for Boltzmann neutrino transport will utilize multi-core sockets a bit differently, but efficiently. The Boltzmann solver has already been parallelized using a domain decomposition in radius. This solver could easily be adapted to run directly on an MPI subcommunicator that spans a socket, or the code can straightforwardly be converted to use a PGAS-language-based decomposition across the socket. As the number of operations necessary for Boltzmann transport solves easily exceeds that for MGFLD by 2 orders of magnitude or more, overheads associated with this approach will not be an issue. Indeed, it will be the floating-point performance of the cores themselves that will determine if these important verification tests can even be attempted.

0.13.4 New Communication Strategies

Although the computational intensity of the hydrodynamics module is dwarfed by that of the neutrino transport and thermonuclear burning components, the parallel performance of the current version of CHIMERA is largely determined by the behavior of the data transposes that occur between hydrodynamics sweeps. This realization becomes especially important when we consider the spatial resolution to be used in production

runs. We expect to perform simulations with CHIMERA with typical angular resolutions of roughly 100×200 angular zones. This will necessitate strong scaling to at least 20,000 sockets, a formidable task utilizing MPI_ALLTOALL, even when the data transpose is not global. In collaboration with N. Wichmann of the Cray Supercomputing Center of Excellence we have begun to investigate the use of PGAS languages (specifically Co-Array Fortran(CAF)) to overlap communication in the data transpose between directional sweeps with computation. As individual "pencils" are evolved by the hydrodynamics module they can be "shipped" into the data transpose while the next one-dimensional sweep is performed. This strategy eliminates synchronization penalties associated with load imbalance during the dimensional sweeps. This overlap cannot occur during the thermonuclear burning/neutrino transport "loaded" directional sweep, but the additional work performed in these components also limits the need to overlap computation and communication in this stage. The success of this development will depend on the availability of a reliable CAF (or, alternatively, UPC) compiler on the target architectures. It will possible to mitigate against the risk of poor or absent PGAS implementations by using some sort of remote memory access model (ARMCI) to accomplish the necessary communication. However, programming this implementation would be considerably less straightforward. Using modern remote memory access libraries is relatively simple, but the nature of CAF and other PGAS languages provides us with a marked degree of expressiveness for his problem.

0.13.5 Parallel I/O & Analysis Tools

The prodigious amount of data from CHIMERA simulations at the petascale will demand robust solutions for parallel I/O. Given the long runtimes for supernova simulation (we anticipate total wallclock times exceeding 2 weeks, even on sustained petaflop platforms), defensive checkpointing is essential. As the thermonuclear kinetics and the neutrino transport modules introduce a large number of additional (compared to pure hydrodynamics) degrees of freedom at each grid point (tens of nuclear species, tens of neutrino energy groups), the memory images for each checkpoint are a large fraction of the total memory used to perform the simulation, approaching 20% of the memory footprint (or more for MGBT). Therefore, I/O performance is essential to ensure that a preponderance of the overall runtime is not taken up by simply writing restart files.

From a data management perspective, it is highly advantageous to reduce the number of files produced in each simulation as much as possible. Though methods and strategies exist to hide the proliferation of individual inodes produced by per-process I/O, we believe the exploratory nature of our data analysis needs ultimately requires a manageable number of files to be produced and archived for future comparisons. Therefore, collective I/O to fewer files is required. It should be noted that this aim is somewhat antithetical to the first point, as the most effective bandwidth is almost always realized by per-process I/O.

To strike an effective compromise between these two divergent aims, we have begun to develop a modular I/O interface for CHIMERA that can make use of evolving I/O methodologies, ranging from per-process to truly globally collective, along with an intermediate solution where files are produced by a set of writer processes after being collected from a subset of compute processes. The per-process I/O and collective writer

modules currently produce raw binary files. Our global collective I/O module currently produces HDF5 files. In none of these cases are we wedded to a particular file format: our I/O infrastructure will allow us to produce pNetCDF files, serial NetCDF files, or raw binary global files via MPI-IO just as easily. We view this range of I/O strategies as risk mitigation: our preference is to produce platform-independent HDF5 files, one per simulation data dump, which can more easily be stored, moved, and manipulated. The collective writes and per-process I/O strategies are "fall-back" positions, as the performance and robustness of collective I/O over hundreds of thousands of processes is, to be conservative, a topic of active research.

In each case, we also are developing a set of data readers for each format for visualization packages like ViSIT and EnSight. We will work closely with our collaborators in the Visualization Task within the Scientific Computing Group at the ORNL Leadership Computing facility to arrive at an optimal set of tools to extract the information required for scientific insight from our data sets.

0.13.6 Data Movement and Workflow Management

The multiphysics nature of core-collapse supernova simulation results in large volumes of high-dimensional data that are strongly correlated. The challenges of managing the sheer volume of data are severe. Even more important is the need to glean insight from the data themselves to ascertain causal relationships between phenomena (the influence of neutrino heating and thermonuclear energy generation on local mater dynamics) and, hopefully, to identify important or pathological features in simulations. The CHIMERA team has a long history of collaboration with a variety of computer science efforts concerned with data management and storage. The vagaries associated with the specific configuration of online and offline storage, the level of network connectivity, and level of integration of petascale platforms with their surrounding infrastructure make dependence on any one set of tools chosen in advance foolish and overly constraining. Instead, we will continue close collaboration with several groups dedicated to producing tools that will (a) be robust, production-ready pieces of software and (b) provide added value to our current mode and future modes of data analysis and movement. Current computer science collaborations include:

- Designing data flow tools to integrate simulation and data analysis and provide data provenance (M. Vouk, NCSU; funded through DOE)

- Using wide area distributed high-speed storage with Logistical Networking tools (REDDiNET collaboration; funded through NSF/MRI)

- Developing and implementing tools for monitoring and steering large-scale simulations (N. Rao, ORNL; funded through UT-Battelle)

- Investigating new approaches to disk I/O that adapt to a users access patterns (X. Ma, NCSU; funded through NSF)

0.13.7 Spatial Grid

CHIMERA uses a spherical polar grid, the most intuitive and straightforward way to implement the ray-by-ray approximation. This allows a quick upgrade path from the existing 2D axisymmetric code, but the coordinate singularity of the polar axis presents a significant problem. In addition to placing a severe Courant limit on the time step, advection errors near the axis are extremely hard to eliminate (c.f. [18]). One method to avoid artifacts of this coordinate singularity is to implement an overset grid [see the Yin-Yang grid described in [81]] in CHIMERA. This scheme uses two spherical polar grids aligned by 90 degrees with respect to each other. The polar regions of one grid are replaced by the equatorial region of the other, with overlapping regions used to implement boundary conditions. We have begun testing a 2D version of this scheme (on a spherical surface), and expect to have a working 3D hydro version soon. We are still left with a singularity at the coordinate origin, but from our experience with the current 2D axisymmetric simulations, we do not expect this to present an intractable problem.

0.13.8 Three-Dimensional Gravity

We have recently implemented the spectral Poisson solver of [117] in CHIMERA. We have also begun to extend the implementation of this solver to three dimensions. This approach has the distinct advantage of requiring only a single global sum across processors. Furthermore, the computational work per zone scales only as $N_r(L+1)^2$, where N_r is the number of zones in the radial direction and L is the order of the highest spherical harmonics used in the expansion of the approximate potential. As a result, this algorithm should prove both efficient and highly scalable. The single global sum, though modest compared to the requirements of many Poisson solvers, does present the possibility of a bottleneck. If the communication required proves to impair the scalability of the method, the computational intensity presented by the thermonuclear burning and neutrino transport modules provide ample work to be overlapped with communication via the use of non-blocking MPI calls.

The most recent version of the code has also been augmented with an approximation to general relativity known as a pseudo-Newtonian potential [98] wherein the classical spherically averaged potential is replaced by a spherically averaged general relativistic potential. This approximation captures most of the post-Newtonian effects in core-collapse scenarios and the gravitational waves produced using this technique have been shown to be remarkably similar to those coming from fully relativistic treatments [49, 50, 95]. However, given the exquisite energy balance inherent in the problem (the explosion energy of 10^{51} erg is roughly 1% of the gravitational binding energy of the nascent neutron star), core collapse simulations require precise conservation of energy and a more accurate treatment of the gravitational fields will be needed. One approximation – the Conformally Flat Approximation (CFA) [152, 153] – has shown superb quality in supernova simulation [49, 50, 39]. The CFA approximates the full solution to the Einstein equations by a series of flat spacetime snapshots that are deformed by a conformal factor that can vary in space and time.

We plan to implement the CFA in CHIMERA in the framework of Yin-Yang grids.

The success of this project relies on the development of highly scalable elliptic solvers fine-tuned to the requirements of general relativistic simulations. These solvers will have applicability in other areas of numerical research where the presence of compact objects (such as black holes and neutron stars) makes the framework of general relativity a necessity. CHIMERA team member Marronetti has been part of the group that developed the CFA technique (Wilson, Mathews & Marronetti 1996) and also a member of the group that first introduced the ”moving puncture” method (Campanelli et al. 2006).

Later, we will implement a more fundamental formalism for general relativistic gravity in CHIMERA known as BSSN [141, 11] that fully solves the Einstein field equations without approximations. As opposed to CFA, which relies on the solution of elliptic equations, BSSN evolves seventeen gravitational fields (four of which are gauge fields) using hyperbolic equations much like standard hydrodynamical algorithms. The implementation and optimization of formulations such as BSSN in highly scalable codes have become top priorities for many groups. In particlar, in the past two years the implementation of BSSN has recently gained widespread use because of its ability to enable binary black hole simulations [36, 8], eliminating insidious numerical instabilities that plagued essentially all codes to that point.

0.13.9 Magnetohydrodynamics

The replacement of the hydrodynamics module in CHIMERA with a module capable of solving the equations of magnetohydrodynamics (MHD) is development task that has not been directly delineated here, but is a highly desirable enhancement to the code. Such a capability will provide CHIMERA with a complete set of known physics. All currently known important phenomena could then be incorporated in CHIMERA simulations including the effects of stellar magnetic fields. This replacement should prove straightforward, but will require some considerable algorithmic work. Given our long experience with finite volume schemes, our preference would be to extend our current hydrodynamics solver to MHD, with a PPM-capable MHD scheme. Also, we can benefit from the publicly available code ATHENA [64]. ATHENA utilizes an unsplit Godunov method with Constrained Transport to solve the ideal MHD equations. Incorporating ATHENA as a CHIMERA module would leverage the considerable effort of the ATHENA developers over many years while preserving much of the code architecture of CHIMERA, as the replacement would not change the computational intensity of the hydrodynamics module as drastically as other choices (divergence cleaning involving a global Poisson solve).

0.14 Timetable

The timeline of 4 years is designed to carry development through the first year of operation of an anticipated sustained petaflop system. The final year will include testing of the integrated code and the performance of production runs. Work on the Yin-Yang grid will be complete by the end of FY10. The CFA solver development will occur in concert with the Yin-Yang development. The full integration of the new XNET module

Task	Y1	Y2	Y3	Y4
Poisson solver development	■	■		
Development of Yin-Yang grid	■	■	■	
QSE for thermonuclear network module	■	■		
Multi-core version of thermonuclear network module		■	■	■
Scalability of elliptic solvers	■	■		
Hyperbolic gravity solver			■	■
Multi-core preconditioner for MGFLD	■	■		
Boltzmann transport module		■	■	■
Parallel I/O modules	■	■		
New data transpose (PGAS)			■	■
Data analysis and movement	■	■	■	■
MHD module			■	■

Table 5: Timetable of tasks [Y1 (FY08) through Y4 (FY11)].

depends on the development of the QSE techniques, but is anticipated to be complete by the end of FY11. The first experiments with multi-core aware transport solves will begin by the end of FY08 and will continue thereafter. The I/O modules will also be continuously updated, with the first deployed examples available by the end of FY08. The work required to implement the MGBT solver as an alternative transport module will begin in FY09, with verification testing occurring in FY11 (assuming adequate single-core performance). The development of an MHD module will begin near the end of FY09, continuing through FY11.

0.15 Results from Ongoing 2D Simulations

We have used CHIMERA to perform two-dimensional simulations of core collapse supernovae [27]. The outcomes in these simulations are promising and have forged a deeper and somewhat new understanding of the supernova mechanism, as we now discuss. We find, as others do, that initially the shock wave, generated at core bounce, stalls and becomes a quasi-stationary accretion shock after propagating out to less than 200 km. 200–500 ms then elapse (the exact figure depending on the progenitor and the stochastic nature of the developing flow). During this time, large-scale convective motions develop between the gain radius (the boundary separating the region suffering net neutrino cooling below from net neutrino heating above) and the shock. Distinct high-entropy plumes, which can be seen by 50 ms after bounce, grow and become separated by narrower, low-entropy downflows. The rising high-entropy plumes begin to push the shock outward, causing local dome-like distortions. As the evolution continues, these plumes merge and grow, distorting the shock even more. By 140 ms for an 11.2 $M_\odot$, or 200 ms for a 15 $M_\odot$ or 20 $M_\odot$ progenitor, two or three plumes dominate the flow. At this time the shock, now showing large dome-like distortions, deflects the incident matter flux when the latter is not normal to the shock interface. This is particularly evident in the vicinity of the shock depressions, where the incoming flow

is funneled into low-entropy downflows. Also, and very significantly, at this time the shock begins to exhibit global distortions of a quasi-oscillatory time-dependent character, which we attribute to the $l = 1$ SASI mode. The shock continues to undulate in a global low-order mode until the oxygen layer, which has been infalling with the matter outside the shock, begins to impinge on one of the extended lobes of the shock. This happens by 180 ms post bounce for an 11.2 $M_\odot$ progenitor, and by 300–500 ms for the 15 or 20 $M_\odot$ progenitor. When it does, the energy generated by the oxygen burning of the shock-heated material causes the shock undulations to rapidly grow and a highly asymmetric explosion to occur in the 11 $M_\odot$ case. The 15 and 20 $M_\odot$ cases are ongoing, and the final outcomes are yet to be determined in these cases. A critical time scale between the stall of the shock and its subsequent rejuvenation is needed in order to explain details of the element synthesis, neutron star masses, and for low-order convective-like modes to have time to develop. The latter are needed to explain many of the supernova observables stemming from large-scale asymmetries. Models which explode too quickly [79, 57, 58, 59, 60, 86] do not develop low-mode distortions. In our proposed supernova scenario, as briefly described above, this critical time scale is naturally provided by the time required for the oxygen layer to advect down to the shock. Until that time has elapsed, neutrino energy deposition supports the shock at a mean radius of several hundred kilometers, but is not able to revive it. Because the neutrino energy deposition rate is highest near the gain radius, tapering off toward the shock with the radial dilution of the neutrino energy density, the region between the gain radius and the shock is rendered Ledoux unstable. Thus, the neutrinos support the shock and break the spherical symmetry by generating a Ledoux unstable region, thereby setting the stage for the explosion. The oxygen burning, which commences when the oxygen layer is advected through the shock, provides an additional source of energy. Unlike the neutrino energy deposition, however, the energy released by oxygen burning occurs in the critical region immediately behind the shock. This is sufficient to initiate a shock revival.

The results described above are exciting and have led to a revised thinking about the fundamental shock reheating paradigm. The shock is not revived while still in the iron core. Rather, it is revived when it reaches the oxygen layer. Moreover, the fundamental explosion time scale is longer than previously imagined, and the role of nuclear burning had been underestimated in the past. Nonetheless, these simulations were performed only in two spatial dimensions, and they must now be performed in three spatial dimensions. Of course, the two-dimensional models will help us understand the outcomes in three-dimensional simulations, and from a practical standpoint, having ongoing two-dimensional multi-physics simulations positions us well to move to three dimensions quickly.

0.16 Planned 3D Simulations: Core Collapse Supernova Mechanism and Observables

Our primary goal is to perform the first three-dimensional core collapse supernova simulations with multifrequency neutrino transport. Associated with this goal are two

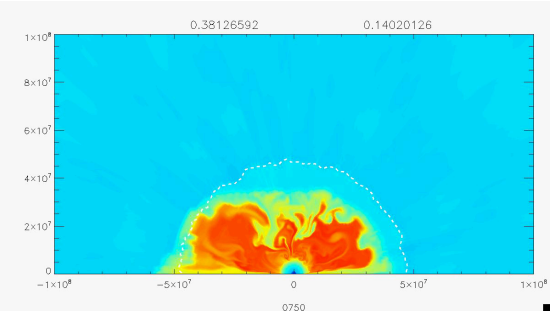


Figure 5: An explosion is initiated in a 20 Solar mass progenitor. The rising, high-entropy (red) convective plumes below the shock are evident. The global distortion of the shock, forming a bipolar explosion, is also evident and is due to the stationary accretion shock instability (SASI). The oxygen layer is delineated by the dashed white line.

subgoals: (1) to ascertain the core collapse supernova mechanism and (2) to predict supernova observables, particularly as they relate to NASA's mission science.

Regarding subgoal (1), one of the most fundamental questions that remains unanswered is whether or not core collapse supernovae are radiatively driven (by neutrinos), magnetically driven, or both [see, for example, [104]]. Our simulations will not include magnetic fields in the course of this proposed work (although in the longer term they will), but the outcomes in our proposed three-dimensional simulations with all but magnetic fields included, particularly given the implementation of multifrequency neutrino transport, will shed light on the role of magnetic fields as well. For example, if robust explosions are obtained in all of our models, for a range of stellar progenitors, we would not conclude that magnetic fields are a necessary ingredient in the core collapse supernova explosion mechanism. On the other hand, if the opposite were true, this would suggest that either magnetic fields are needed or there is some yet-to-be-discovered essential physics component that has not been included in our (or any) models. Of course, even if we do obtain robust explosions for a range of progenitors, without magnetic fields, we will ultimately need to explore the impact of magnetic fields in these models as well.

Year	Progenitor	Grid, Gravity
1 (FY08)	Nonrotating 11 $M_\odot$	θ-Axis Excision, Newtonian with Spherical GR Corrections
2	Nonrotating 15 $M_\odot$	θ-Axis Excision, Newtonian with Spherical GR Corrections
2	Rotating 15 $M_\odot$	θ-Axis Excision, Newtonian with Spherical GR Corrections
2	Nonrotating 20 $M_\odot$	θ-Axis Excision, Newtonian with Spherical GR Corrections
3	Rotating 15 $M_\odot$	Yin-Yang, CFA
3	Nonrotating 25 $M_\odot$	Yin-Yang, CFA
3	Rotating 25 $M_\odot$	Yin-Yang, CFA

The table above lists the sequence of models that will be developed. Our simulations will begin with different progenitors—e.g., 11, 15, and 25 $M_\odot$ progenitors—and will consider both nonrotating and rotating cases. Two-dimensional simulations beginning with an 11 $M_\odot$ nonrotating progenitor exhibit explosions on a fairly rapid time scale. For this progenitor, the oxygen layer is reached at $\sim$180 ms after bounce, whereas this does not occur until $\sim$ 400–500 ms after bounce for more massive nonrotating progenitors. Moreover, given the stellar core profiles in this case, explosion is likely in three dimensions (it already occurs in two). Thus, our first three-dimensional multi-physics simulation will begin with an 11 $M_\odot$ nonrotating progenitor. We will extend this to the more massive 15 $M_\odot$ progenitor with and without rotation and to the 20 $M_\odot$ progenitor without rotation. We will further extend our models to more massive progenitors and consider nonrotating and rotating 25 $M_\odot$ models. The extension to 25 $M_\odot$ progenitors will be accompanied by the extension of our spherically-averaged treatment of general relativity to the more sophisticated conformally flat approximation, given the increased gravitational fields in this case. In addition, we will deploy the Yin-Yang grid. Thus, a repeat of the 15 $M_\odot$ model but with the Yin-Yang grid and the CFA will allow a comparison of the effects of the choice of grid and approximation to general relativity on the simulation outcomes. The number of simulations performed in a given year is limited by the CPU requirements. For example, three-dimensional simulations at 3-degree resolution in θ and ϕ (64×128 resolution) will require the use of 8,192 processors for a period of two weeks. This amounts to 2.75 million processor–hours per run. [We would also perform runs at 1-degree resolution (128×256 resolution) for a window of postbounce time to explore the impact of resolution on the simulation results.]

A number of other questions will be addressed by our 3D simulations: (i) What is the nature of neutrino-driven convection in three dimensions and its role in the supernova mechanism? Past two-dimensional simulations have demonstrated the importance of neutrino-driven convection during neutrino shock reheating [69, 33, 79, 106, 29, 27]. What will neutrino-driven convection look like in three dimensions without the imposition of axisymmetry? This was first explored by [60] in simulations that deployed gray neutrino transport. The development of neutrino-driven convection depends on the neutrino shock reheating, which determines the shock's position relative to the gain radius and, thus, the size of the convective volume and its time dependence. In turn, the neutrino heating is very sensitive to the neutrino spectra. Thus, to capture the development of neutrino-driven convection below the shock we must advance to three dimensions *and* use multifrequency neutrino transport. (ii) Are there fluid instabilities in the proto-neutron star (below the neutrinospheres) that lead to vigorous convective, or other, flows that in turn affect the neutrino luminosities in a significant way? In past two-dimensional simulations that deployed multifrequency neutrino transport, the extent and impact of such instabilities has been varied [29, 27], and the extension to three dimensions may fundamentally alter the outcomes in either case. Of particular note is the following question: Does the subclass of proto-neutron star instabilities known as neutron fingers exist and, if so, does it have a significant impact on the explosion dynamics? The LLNL supernova models of Wilson and collaborators require neutron fingers to obtain explosions [154]. Neutron fingers are a "doubly diffusive" instability, relying on competing gradients for their development; thus, they take longer to develop.

The existence of neutron fingers has yet to be confirmed or denied definitively, and the now longer run times common in supernova models once again beg the question. (iii) What is the role of rotation in the supernova mechanism and in defining a dynamical, physical axis in a simulation that has no imposed symmetries? Rotation has played a significant role in past two-dimensional models with multifrequency neutrino transport [29, 139], but two-dimensional simulations are numerically severely constrained. (iv) How will the stationary accretion shock instability develop in three dimensions in more complete multi-physics simulations and what role will it play in the explosion mechanism? The existence of the SASI in three dimensions was confirmed by a series of hydrodynamics simulations (Blondin, 2005). In addition to showing that the *l =1* mode was still the dominant *linear* mode in 3D, these simulations found that the transition to the nonlinear phase of the SASI was typically marked by a loss of axisymmetry and the formation of a very robust *m =1* mode. Moreover, the *m=1* mode led to the accretion of significant angular momentum onto the proto-neutron star, perhaps sufficient to explain the rapid rotation of young radio pulsars (Blondin & Mezzacappa 2007). Thus, the move to three dimensions will admit new degrees of freedom and will fundamentally alter the SASI's development and the resultant shock dynamics. How long will it take for the SASI to develop? How long will it operate in the stellar core? When will its impact end? Answers to these questions will require models that are both three-dimensional and multi-physics. In turn, answers to these questions will determine the SASI's role in defining important supernova observables such as remnant neutron star spin.

Regarding subgoal (2) above, based on our simulation data we will make predictions for (i) supernova explosion energies, (ii) neutron star masses, (iii) neutron star kick velocities, (iv) neutron star spins, and (v) supernova nucleosynthesis. An estimate of the explosion energy is obtained by adding up the total energy (gravitational, internal, and kinetic) of the ejecta moving in the positive radial direction. We have ascertained in two dimensions that simulations must be carried out to at least 0.7 to 0.8 s post bounce in order to obtain realistic estimates of the explosion energy. The neutron star masses, modulo appreciable fallback, are straightforward to compute. In two dimensions, we currently obtain neutron star masses of about 1.42 $M_\odot$ for a non-rotating 11.2 $M_\odot$ progenitor, and 1.54 $M_\odot$ for a 15 $M_\odot$ progenitor. To obtain the neutron star kick velocities, CHIMERA has the option of either computing the total momentum of the material plus neutrinos outside the radius of the neutron star (taken as the radius at which the density is $10^{11}\mathrm{gcm}^{-3}$) and attributing an equal and opposite momentum to the neutron star, or performing a Galilean transformation after each time step wherein the neutron star remains at the center of the grid and the ejecta move coherently in the opposite direction of the neutron star's recoil velocity.

Likely the most sensitive test of a supernova model is the nucleosynthesis that results, and of this the innermost ejecta are most directly affected by the explosion mechanism—i.e., the yields of Fe group nuclei. Most recent calculations of the nucleosynthesis resulting from the core collapse of a massive star employ an induced piston or thermal bomb to simulate the explosion [157, 145, 122, 119, 132, 40, 41, 148, 149]. But this treatment fails to account for the effect of neutrino interactions. A quantity indispensable to correctly describe nucleosynthesis in the innermost ejecta is the electron fraction, $Y_e =< Z/A >$, in the layers undergoing explosive Si burning. This Y_e is set

by the weak interactions in the explosively burning layers—i.e., electron and positron captures, nuclear decays, and neutrino and antineutrino captures [145, 120, 55, 56]. To determine the elemental abundance yields produced by supernova models, we propose to distribute Lagrangian tracers throughout the matter that will be ejected. Each of these tracers will provide a (T, Y_e)-trajectory of the mass element to which the tracer is attached. This tracer data will then be used to post-process the mass element with the full nuclear network to obtain the isotopic abundance yields. Later, we will perform fully self-consistent calculations of the abundance yields by using either the full network in two- (three-) dimensional core collapse simulations with CHIMERA, or a reasonable approximation of the network if the use of the full network in 3D is prohibitive. These simulations will not only be extremely important tests of our models, but will provide a much needed link to the rapidly expanding field of galactic chemical evolution, which is being energized by the large amount of recent abundance observations from low-metallicity stars [3, 4, 143, 38, 75, 53].

It is well known that stellar core collapse, if only slightly aspherical, will lead to copious gravitational wave emission. The gravitational waveforms and associated spectra bear the direct stamp of the hydrodynamics and rotation of the core and speak volumes about internal supernova evolution. We currently track the time evolution of the quadrupole moment of the mass distribution in our 3D hydrodynamics simulations to provide predictions of the gravitational wave signatures. Our goal is to compute the gravitational wave signatures of our more complete, multi-physics models and thereby supply important and unique data to currently employed gravitational wave detectors like LIGO, VIRGO, GEO600, and TAMA. Later, we will take this one step further by developing a GR solver using the conformally flat approximation [153] and incorporating this into the version of CHIMERA that will deploy a Yin-Yang grid.

GTC-S Discussion

0.17 GTC-S: Gyrokinetic Simulation of Global Turbulent Transport in Fusion Plasmas

GTC-S is a general geometry gyrokinetic particle simulation code for studying plasma turbulence relevant to present-day tokamak experiments. It is a descendant of the original GTC code. In this new version of the code, we have incorporated relevant physics modules such as the comprehensive influence of non-circular cross section, realistic plasma profiles, plasma rotation, neoclassical (equilibrium) electric fields, and Coulomb collisions. These are most important ingredients for understanding anomalous transport in modern tokamak experiments in the US, such as NSTX at Princeton Plasma Physics Laboratory and DIIID at General Atomics. Most of all, we are preparing ourselves for simulating the future experiments in ITER, an international project with participation from 7 countries including the US. Toward that goal, we have been working on the incorporation of the electron dynamics and the electromagnetic effects into GTC-S through the SciDAC Gyrokinetic Particle Simulation Center (GPSC) and the OASCR Multi-Scale Gyrokinetics (MSG) Project. We welcome the opportunity that affords to us through the participation of the Joule Applications to improve the performance of GTC-S. We hope this exercise would be a stepping stone for us in joining the petascale campaign at ORNL in the not too distant future. In the physics front, we also would like to seize this opportunity to carry out simulations for electron temperature gradient (ETG) turbulence for NSTX, which remains a possible candidate for the electron thermal transport observed in the experiments.

0.17.1 Introduction

Understanding turbulence and associated transport in toroidal plasmas[158, 159, 160] is one of the key issues in magnetic fusion research. In the past decade, as computer resources rapidly increased and advanced numerical algorithms were developed, significant progress was made for this long-standing complicated issue through computer simulation based on various approaches.[161]-[172] Among them the first-principles based gyrokinetic particle approach[173] has been widely employed. Simulation studies carried out with the gyrokinetic toroidal code GTC[165] have been among the most productive. GTC was originally developed to focus on fundamental nonlinear turbulence physics. It is a full-torus global code using a global field-line-following mesh

and a real space field solver. Global turbulence simulations for toroidal plasmas are highly demanding for the following reasons: (i) the turbulence-generated zonal flow contains radial scales as large as the system size, even though the turbulence itself is on the much smaller scale of the gyroradius; (ii) the equilibrium $\mathbf{E} \times \mathbf{B}$ shear flow, which also plays an important role in determining turbulence levels, typically has the large scale size of the plasma minor radius; (iii) turbulence spreading to the linearly stable zone results in nonlocal transport, which is a truly global phenomenon. To pose the simplest problem while keeping the important global physics properties, a simplified model was utilized in the previous simulations, with simple magnetic geometry with a large aspect ratio circular concentric cross section and neglect of the effect of radial variation of pressure. This proved to be an effective means of gaining key insights into the complexity of the toroidal turbulence system. As a result, the previous simulation studies have led to a number of important understandings with regard to zonal flow effects, transport scalings with collisionality and device size, turbulence spreading, etc.[165],[174]-[178] While such a simplified model is a useful tool to separate and clarify fundamental physics issues, more realistic features are needed as the research moves forward. Particularly for simulating turbulence phenomena in tokamak experiments, a more comprehensive model is needed which consistently incorporates the influence of general geometry, realistic plasma profiles, plasma flow, neoclassical equilibrium, Coulomb collisions and other features. Here, we present a new model with emphasis on the general geometry capability implemented based on GTC architecture. This general geometry simulation capability, called GTC-S, has been interfaced with TRANSP,[179] a widely used experimental data analysis software tool for specifying experimental plasma profiles of temperature, density and toroidal angular velocity, and also with various numerical magnetohydrodynamic (MHD) equilibrium codes, including the JSOLVER[180] and ESC[181] codes. The new capability has been applied successfully to shaped plasmas with DIII-D[182] geometry to examine the ion temperature gradient (ITG) turbulence spreading phenomena and the global turbulence evolution process. We have also investigated the nonlinear interplay between turbulence and zonal flow, which is essentially a local process. As an attempt to understand the nonlinear zonal flow dissipation, which is needed to saturate zonal flow and to explain the mutual self-regulation observed in collisionless simulations, we have studied the process of zonal flow driving turbulence. Our simulation results suggest that the coupling of zonal flows to turbulence and associated energy transfers are much weaker, compared to the opposite process in ITG turbulence.

In order to carry out more systematic validation studies of GTC-S against the experimental data, it is time for us to improve the performance of GTC-S. Thus, our participation in the Joule Applications is very timely. With this type of improvement, we can use GTC-S to study ETG turbulence using realistic NSTX parameters. This is a very timely exercise and ETG turbulence is very controversial topic in the magnetic fusion community.

For the next section, we will describe the basic physics principles and the numerical schemes for GTC-S and its similarities and differences in comparison with the original GTC. The numerical issues will be discussed in Sec. III. We will describe the ETG simulations and future plans in Sec. V.

0.18 General Geometry Gyrokinetic Particle Simulation Model

0.18.1 Coordinate system and mesh construction

Magnetic flux coordinates, in which the radial coordinate labels magnetic surfaces, are generally used for toroidal systems and associated with MHD equilibria. Our gyrokinetic simulation in principle can use arbitrary flux coordinates with straight field lines. In the flux coordinates, the global field-line-following mesh, which possesses the highest efficiency by capturing the flute-type character of the drift wave turbulence in toroidal plasmas, can be easily constructed. A preferable flux coordinate can be chosen in terms of different requirements. A symmetric coordinate system in which the toroidal angle φ is chosen to be the azimuthal angle of cylindrical coordinates is preferable in many cases. These coordinates are relatively uniform compared to others which have been previously used, and advantageous for constructing a relatively regular mesh in real space for strongly shaped plasmas. It also facilitates straightforward visualization with the poloidal plane defined with the physical angle φ. The radial coordinate is defined as $r = \sqrt{\psi/\psi_e}$ where ψ and ψ_e are the toroidal flux and its value on the plasma boundary, respectively. This same radial coordinate is widely used in the experimental community.

Because of the flute-type character of drift wave turbulence in toroidal plasmas, with $k_\| \ll k_\perp$, where $k_\|$ and $k_\perp$ are the parallel and perpendicular wave numbers, respectively, GTC uses a field-line-following mesh which shows high efficiency for calculating the turbulent field. Note that the field-line-following mesh does not possess toroidal symmetry. On the other hand, in tokamak geometry, the toroidal symmetry is broken as fluctuations associated with microinstability develop. Each mode nonlinearly sees an asymmetric equilibrium which consists of the MHD equilibrium, which is symmetric, and the fluctuations of other modes, which are asymmetric. In our simulation, the MHD equilibrium is represented by a 2-D symmetric mesh, and the fluctuations are represented using a (toroidally non-symmetric) field-line-following mesh which best represents the nature of the mode structure. For drift wave turbulence, the spatial scale length in the perpendicular direction is generally in correlation with the local gyroradius $\rho_j \propto \sqrt{T_j}$, which may vary substantially from the core to the edge of the plasma. For instance, it is common in National Spherical Torus Experiment (NSTX)[183] plasmas that the ion temperature changes from $\sim keV$ in the core to $\sim 10eV$ near the separatrix region at the plasma edge. Therefore, for a global simulation, which includes the entire radial domain, it is important to use a nonuniform grid with the grid size in the perpendicular direction correlated with the local gyroradius for improved spatial resolution and efficiency. To this end, we re-scale the radial coordinate by defining ρ as follows:

$$\frac{d\rho}{dr} = \sqrt{T_c/T_i(r)}, \tag{245}$$

where T_c is the temperature at a reference radial location. Working with the new coordinate ρ, we use an evenly spaced radial grid, which offers great convenience for frequent operations such as particle sorting, charge deposition, gathering, etc. This allows the grid size in real space to be correlated with the local gyroradius: $\Delta r \sim \sqrt{T_i(r)/T_c}$.

In the poloidal direction, the grid size $\Delta\theta(r)$ is uniform on a flux surface, while varying over different flux surfaces. The grid size $\Delta\theta(r)$ is determined so as to make the poloidal arc length Δl_θ near the mid-plane correlated with ρ_i. An example of such a grid on the poloidal plane $\varphi = 0$ is shown in Figure[0.18.1]. Generally, a two-dimensional mesh on the $\varphi = 0$ plane is set up first. A three-dimensional mesh is constructed by following each (approximate) field line, which starts at a grid point on the $\varphi = 0$ plane and has $\bar{q}(r)\theta - \varphi =$constant, with $\bar{q}$ slightly changed from the usual safety factor $q(r)$ so that the approximate field lines will lead back to one of the grid points on the $\varphi = 0$ plane. Two methods for enforcing toroidal periodicity have been implemented. One method is to map the grid at $\zeta = 0$ to the grid at $\zeta = 2\pi$ using interpolation, which results in some spatial damping. Another method is to allow the grid to slightly depart from the magnetic field lines in order to match the grid points, which requires a chain rule in calculating the parallel derivatives. Interpolation and deposition, which transfer information between particles and grid points, are performed within a "flux-tube" type of cell twisted along the field line rather than a toroidally symmetric cell. This has advantages due to separating the short spatial scale in the perpendicular direction and the long spatial scale in the parallel direction. In the field-line-following mesh system, the number of "toroidal grids" (i.e, the number of poloidal planes) is actually the number of grids in the parallel direction within one toroidal circuit. In our simulation, the number of "toroidal grids" N_t is chosen to give adequate resolution for the parallel structure of the modes. In the perpendicular direction, adequate resolution is achieved by using a much denser mesh on each poloidal plane with a perpendicular grid size of ρ_i scale. The allowed maximum toroidal wave-number is related to both the "toroidal" and poloidal grids. The key point is that the poloidal mesh makes a significant contribution to the resolution in the toroidal direction. This problem was addressed in detailed by B. Scott.[184] Basically, the maximum toroidal mode number which can be resolved is $n = \text{NINT}(m/q) + N_t/2$ for each poloidal mode number m, where NINT(m/q) is the nearest integer to m/q, and $N_t/2$ is the so called "toroidal Nyquist number". Therefore, adequate resolution is also guaranteed in the toroidal direction.

Note that the field-line-following mesh described above is used only for calculating fluctuations. A separate 2-D mesh in the same coordinate system with a uniform grid in both the radial and poloidal directions is used to represent the axisymmetric MHD equilibria numerically. This mesh has a larger grid size, as the equilibrium scale length is much larger than that of the turbulence. Multi-dimensional spline interpolation is used to obtain equilibrium quantities at any spatial location.

The gyrokinetic particles are followed in general flux coordinates using guiding center Lagrangian equations, instead of Hamiltonian equations that require construction of canonical variables[185] which are complicated forms in general geometry and inconvenient to use. The guiding center Lagrangian obtained by Littlejohn has the following normalized form[185, 186]

$$L(\mathbf{x}, \dot{\mathbf{x}}; t) = (\mathbf{A} + \rho_\parallel \mathbf{B}) \cdot \mathbf{v} - H, \tag{246}$$

with the guiding center Hamiltonian $H = \rho_\parallel^2 B^2/2 + \mu B + \Phi$. Here, the magnetic field $\mathbf{B} = \nabla \times \mathbf{A}$, $\rho_\parallel = v_\parallel/B$ is the parallel gyroradius, μ is the magnetic moment, and Φ

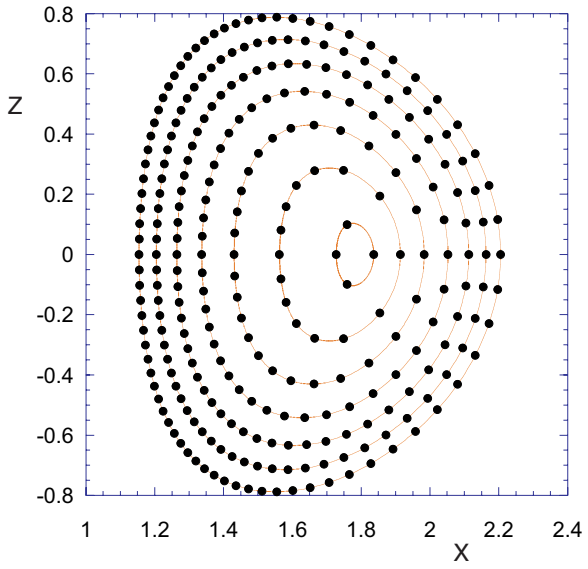


Figure 6: An example of a nonuniform grid on a poloidal plane showing grid size in correlation with local ion gyroradius.

is the electric potential. The independent variables are $\mathbf{x} = (r, \theta, \varphi, \rho_{\|})$. The particle guiding drift motion is governed by the Lagrangian equations

$$\frac{d}{dt}\left(\frac{\partial}{\partial \dot{x}_i} L\right) - \frac{\partial}{\partial x_i} L = 0. \tag{247}$$

The obtained equations for $d\mathbf{x}/dt$ are suitable for any generalized flux coordinates.

0.18.2 Gyrokinetic transformation

One of the important elements of the gyrokinetic formalism ([173],[187]-[191]) is concerned with the transformation of fluctuations between the particle position $\vec{x}$ and the guiding center position $\vec{R}$. The fluctuations, such as the potential ϕ and the ion density δn_i, in the two coordinates are connected by the gyrokinetic transformation which is expressed as follows

$$\bar{\phi}(\vec{R}, \mu) = \frac{1}{2\pi} \int \phi(\vec{x}) \delta(\vec{x} - \vec{R} - \vec{\rho}) d\vec{x} d\Theta, \tag{248}$$

$$\tilde{\phi}(\vec{x}) = \frac{1}{2\pi} \int \bar{\phi}(\vec{R}, \mu) f_{Mi}(\vec{R}, \mu, v_{\|}) \delta(\vec{R} - \vec{x} + \vec{\rho}) d\vec{R} d\mu dv_{\|} d\Theta, \tag{249}$$

$$\bar{\delta n}_i(\vec{x}) = \frac{1}{2\pi} \int \delta f_i(\vec{R}, \mu, v_{\|}) \delta(\vec{R} - \vec{x} + \vec{\rho}) d\vec{R} d\mu dv_{\|} d\Theta, \tag{250}$$

where $\vec{\rho}$ is the gyroradius vector, Θ is the gyrophase, f_{Mi} is assumed to be Maxwellian and δf_i is the perturbed ion distribution function. In a gyrokinetic simulation, the quantities are calculated either in real space or in Fourier space. In real space the transformation is carried out by the four-point averaging scheme. [192] The exact gyro-average is performed on a gyro-plane perpendicular to the magnetic field, with four points evenly spaced on a gyro-orbit. Because the grid points on which fluctuations ϕ and δn_i are calculated are set up on poloidal planes, it is much more convenient to perform the gyro-average on poloidal planes instead of on gyro-planes. In the case of the simple geometry of large aspect ratio circular concentric cross section, the difference between a poloidal plane and a gyro-plane is neglected in doing the gyro-average. An accurate treatment for the gyro-average in general geometry is obtained by taking into account the finite ratio of the poloidal to the total magnetic field B_p/B, which separates the poloidal plane from the gyro-plane. By projection to the poloidal plane, a gyro-orbit becomes an elliptic orbit (Figure[0.18.2]). In the direction of $(\nabla\varphi \times \mathbf{B}) \times \nabla\varphi$, the gyroradius is elongated by a factor of $1/\cos\alpha$ while there is no change in the $\nabla\varphi \times \mathbf{B}$ direction. Here, $\cos\alpha = \mathbf{B} \cdot \nabla\varphi/B|\nabla\varphi| = \psi'/\mathcal{J}B|\nabla\varphi|$, with $\psi' \equiv d\psi/dr$ and the Jacobian $\mathcal{J} = (\nabla r \times \nabla\theta \cdot \nabla\varphi)^{-1} > 0$ (right handed). The four points used for the gyro-average are chosen to be located on the axes $(\nabla\varphi \times \mathbf{B}) \times \nabla\varphi$ and $\nabla\varphi \times \mathbf{B}$. An ion spends approximately the same amount of time on each quarter of the ellipse. To locate the positions of the four points relative to the guiding center, we first calculate the directional derivatives in the two directions which are defined as $dA/dl \equiv \hat{\mathbf{l}} \cdot \nabla A$ for any function A, where $\hat{\mathbf{l}}$ is the unit vector in direction $\mathbf{l}$. In the $\nabla\varphi \times \mathbf{B}$ direction

$$\frac{dr}{dl_1} = \frac{[(g^{r\varphi})^2 - g^{\varphi\varphi}g^{rr}]/q + g^{rr}g^{\theta\varphi} - g^{r\varphi}g^{r\theta}}{\sqrt{g^{\varphi\varphi}B^2/\psi'^2 - 1/\mathcal{J}^2}}, \tag{251}$$

$$\frac{d\theta}{dl_1} = \frac{[g^{\varphi\theta}g^{r\varphi} - g^{\varphi\varphi}g^{\theta\varphi}]/q + g^{r\theta}g^{\theta\varphi} - g^{r\varphi}g^{\theta\theta}}{\sqrt{g^{\varphi\varphi}B^2/\psi'^2 - 1/\mathcal{J}^2}}; \tag{252}$$

in the $(\nabla\zeta \times \vec{B}) \times \nabla\zeta$ direction

$$\frac{dr}{dl_2} = -\frac{g^{r\varphi}}{\mathcal{J}\sqrt{g^{\varphi\varphi}}\sqrt{g^{\varphi\varphi}B^2/\psi'^2 - 1/\mathcal{J}^2}}, \tag{253}$$

$$\frac{d\theta}{dl_2} = \frac{g^{\varphi\varphi}/q - g^{\theta\varphi}}{\mathcal{J}\sqrt{g^{\varphi\varphi}}\sqrt{g^{\varphi\varphi}B^2/\psi'^2 - 1/\mathcal{J}^2}}, \tag{254}$$

where the metric tensor $g^{\alpha\beta} = \nabla\alpha \cdot \nabla\beta$. The four points used for the ion gyro-average are determined by

$$r_j = r_0 + \Delta r_j, \;\; \theta_j = \theta_0 + \Delta\theta_j, \;\; \varphi_j = \varphi_0, \quad j = 1, 2, 3, 4 \tag{255}$$

with

$$\Delta r_{1,2} = \pm\frac{dr}{dl_1}\rho_i, \;\; \Delta\theta_{1,2} = \pm\frac{d\theta}{dl_1}\rho_i, \;\; \Delta r_{3,4} = \pm\frac{dr}{dl_2}\frac{\rho_i}{\cos\alpha}, \;\; \Delta\theta_{3,4} = \pm\frac{d\theta}{dl_2}\frac{\rho_i}{\cos\alpha}. \tag{256}$$

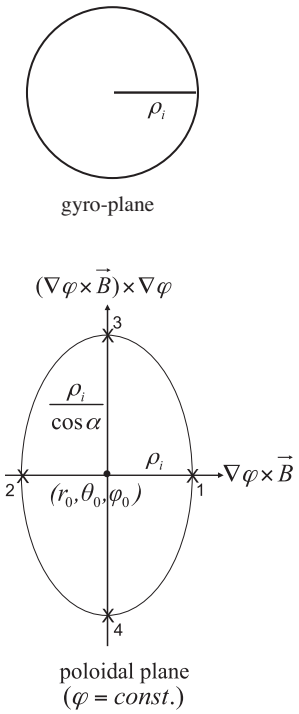


Figure 7: on gyro-orbit.

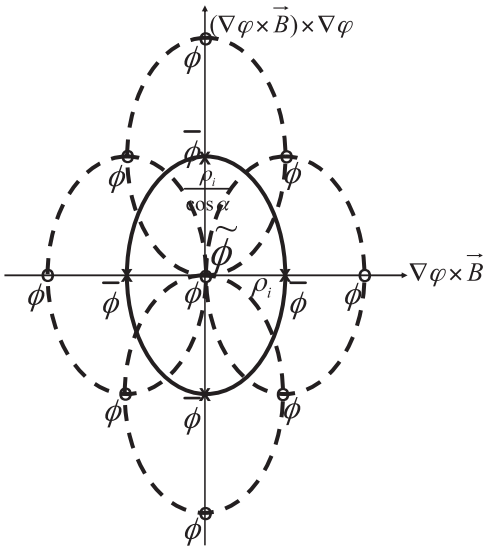


Figure 8: Schematic diagram of double average process for calculating $\tilde{\phi}$ at the center position.

The calculation of the potential $\tilde{\phi}(\vec{x})$ in terms of $\phi(\vec{x})$ involves the double averaging process. [193] Following the above method, we can extend the previous calculation of $\tilde{\phi}(\vec{x})$ to general geometry, taking into account finite B_θ/B. The double average is made along the elliptic orbit projected on the poloidal plane as shown in Figure[gtcs-fig3], where $\tilde{\phi}$ at the center point is the four-point average of $\bar{\phi}$ on the X points, which are other four-point averages of ϕ on the O points. The average over the distribution function is accomplished by carefully sampling different gyroradii. [193]

It is noted that the four-point averaging scheme is accurate for $k_\perp\rho_i \leq 2$ modes. To resolve shorter wavelength modes, we may use more points for the averaging process. An implicit assumption of the 4-point averaging scheme described above is that the equilibrium scale lengths L_p and L_B of the pressure and magnetic field are much larger than the ion gyroradius, which is consistent with the gyrokinetic ordering.

0.18.3 Basic equations

The gyrokinetic particle distribution is expressed as $f = f_0 + \delta f$. Here we separate the turbulence perturbation δf from the equilibrium distribution f_0. In the electrostatic limit the ion gyrokinetic equation for δf_i, with μ and $v_\parallel$ as independent velocity variables, is

$$\frac{\partial \delta f_i}{\partial t} + (v_\parallel \hat{b} + v_{\vec{E}_0} + v_{\vec{E}} + v_{\vec{d}}) \cdot \nabla \delta f_i - \hat{b^*} \cdot \nabla(\mu B + \frac{e}{m_i}\Phi_0 + \frac{e}{m_i}\bar{\phi})\frac{\partial \delta f_i}{\partial v_\parallel}$$

$$= -v_{\vec{E}} \cdot \nabla f_0 + \hat{b^*} \cdot \nabla(\frac{e}{m_i}\bar{\phi})\frac{\partial f_0}{\partial v_\parallel} + C_i^l(\delta f_i). \quad (257)$$

Here $v_{E_0}^{\vec{}}$ and $v_E^{\vec{}}$ are the $\mathbf{E}\times\mathbf{B}$ drift velocities corresponding to the equilibrium potential Φ_0 and the fluctuation potential $\bar{\phi}$ respectively, $v_d^{\vec{}}$ is the ∇B drift velocity, $\hat{\mathbf{b}}^* = \hat{\mathbf{b}} + \rho_\parallel \hat{\mathbf{b}}\times(\hat{\mathbf{b}}\cdot\nabla\hat{\mathbf{b}})$ with $\hat{\mathbf{b}} = \mathbf{B}/B$, and C_i^l is the linearized Coulomb collision operator. Note that a parallel velocity nonlinearity term which is one order higher[30] is retained in the equation. This term is required in order to maintain energy conservation. [188, 189] The equilibrium distribution function f_0 is determined by the neoclassical dynamics and obeys

$$\frac{\partial f_0}{\partial t} + (v_\parallel \hat{b} + v_{E_0}^{\vec{}} + v_d^{\vec{}})\cdot\nabla f_0 - \hat{b^*}\cdot\nabla(\mu B + \frac{e}{m_i}\Phi_0)\frac{\partial f_0}{\partial v_\parallel} = C_i(f_0, f_0). \quad (258)$$

The lowest order solution of Eq.14 is a shifted Maxwellian consistent with plasma rotation:

$$f_0 = f_{SM} = n(r,\theta)(\frac{m_i}{2\pi T_i})^{3/2} e^{-\frac{m_i}{T_i}[\frac{1}{2}(v_\parallel - U_i)^2 + \mu B]}, \quad (259)$$

where the parallel flow velocity U_i is associated with the toroidal rotation by $U_i = I\omega_t/B$ with ω_t the toroidal angular velocity and I the toroidal current, and $n(r,\theta)$ is the ion density $n(r,\theta) = N(r)e^{\frac{m_i U_i^2}{2T_i} - \frac{e\tilde{\Phi}_0}{T_i}}$. The total equilibrium potential consists of two parts $\Phi_0 = \langle\Phi_0\rangle + \tilde{\Phi}_0$. Here, $\langle\rangle$ denotes a flux surface average. The poloidally varying component $\tilde{\Phi}_0$ can be generated by the centrifugal force which drives charge separation on a magnetic surface in strongly rotating plasmas.[194] Generally the radial potential $\langle\Phi_0\rangle$ is dominant. The equilibrium radial electric field can be calculated from a first-principles based particle simulation of neoclassical dynamics with important finite orbit effects, [195, 196] or obtained by direct experimental measurement if available. For equilibrium toroidal plasmas, a shifted Maxwellian with either model or experimental profiles of $\langle n(r,\theta)\rangle$, $T_i(r)$ and $\omega_t(r)$ is prescribed for the ions. The electron dynamics is described by the drift kinetic equation, neglecting the finite gyroradius effect. The electron guiding center distribution is represented as $f_e = f_{e0} - (e\delta\phi/T_e)f_{e0} + \delta h_e$, with turbulence potential $\delta\phi = \phi - \langle\phi\rangle$. The equilibrium distribution f_{e0} satisfies the electron version of Eq. 14 and can be approximated by a shifted Maxwellian containing a parallel flow, similar to that for the ions. The second term represents the adiabatic electron response to the potential fluctuation due to the fast electron motion. The nonadiabatic electron distribution δh_e is determined by

$$\frac{\partial \delta h_e}{\partial t} + (v_\parallel \hat{b} + v_{E_0}^{\vec{}} + v_E^{\vec{}} + v_d^{\vec{}})\cdot\nabla\delta h_e - \hat{b^*}\cdot\nabla(\mu B + \frac{e}{m_e}\Phi_0 + \frac{e}{m_e}\phi)\frac{\partial \delta h_e}{\partial v_\parallel}$$

$$= -v_E^{\vec{}}\cdot\nabla f_{e0} + \hat{b^*}\cdot\nabla(\frac{e}{m_e}\phi)\frac{\partial f_{e0}}{\partial v_\parallel} + \frac{e}{T_e}\frac{\partial\delta\phi}{\partial t}f_{e0} + (v_\parallel\hat{b} + v_{E_0}^{\vec{}} + v_E^{\vec{}} + v_d^{\vec{}})\cdot\nabla\left(\frac{e\delta\phi}{T_e}\right)f_{e0} + C_e^l(\delta h_e). \quad (260)$$

Again, the parallel velocity nonlinearity is retained here for δh_e. At present the trapped electron dynamics described by Eq. 16 is treated as a higher order correction to the adiabatic response via a hybrid model [197, 198] in the electrostatic limit. To include full electron dynamics, we will use the split-weight scheme [199, 200] to solve Eq. 16.

The electrostatic fluctuation potential is divided into a turbulence part plus a zonal flow part: $\phi = \delta\phi + \phi_{00}$ with $\phi_{00} \equiv \langle\phi\rangle$. This expression emphasizes the critical role

of turbulence-generated zonal flow in determining the turbulence and the associated transport level. For the turbulence potential, the gyrokinetic Poisson equation [173] becomes

$$\left(1+\frac{T_i}{T_e}\right)\frac{e\delta\phi}{T_i}-\frac{e\widetilde{\delta\phi}}{T_i}=\frac{\delta\bar{n}_i-\langle\delta\bar{n}_i\rangle}{n_0}-\frac{\delta n_e^{(1)}-\langle\delta n_e^{(1)}\rangle}{n_0}, \tag{261}$$

where $\delta\bar{n}_i=\int d^3v\delta f_i$ is the ion fluctuation density of guiding centers and $\delta n_e^{(1)}=\int d^3v\delta h_e$ is the nonadiabatic density of electrons. Because the zonal flow has a larger spatial scale than the turbulence fluctuations, it is advantageous to solve for it separately in our simulations. The generalized equation for zonal flow in shaped geometry is obtained as:

$$\frac{1}{\mathcal{V}_r'}\frac{d}{dr}\left[\frac{d\phi_{00}}{dr}\mathcal{V}_r'\langle g^{rr}\rangle\right]=\frac{1}{\mathcal{V}_r'}\frac{d}{dr}\left\{\frac{d}{dr}\left[\frac{T_i}{e}\left(\frac{\langle\delta\bar{n}_i\rangle}{n_0}-\frac{\langle\delta n_e^{(1)}\rangle}{n_0}\right)\right]\mathcal{V}_r'\langle g^{rr}\rangle\right\}-\langle\frac{1}{\rho_i^2}\rangle\frac{T_i}{e}\left(\frac{\langle\delta\bar{n}_i\rangle}{n_0}-\frac{\langle\delta n_e^{(1)}\rangle}{n_0}\right), \tag{262}$$

where $\mathcal{V}_r'=\oint d\theta d\varphi\mathcal{J}$. In Eq. 18 we use the Pade approximation $\Gamma_0(b)\equiv I_0(b)e^{-b}\approx 1/(1+b)$ with I_0 the modified Bessel function and $b=(k_\perp\rho_i)^2$, and $\left\langle\tilde{\phi}\right\rangle\approx\widetilde{\langle\phi\rangle}$. The later approximation is not well justified for low aspect ratio geometry. A generalized field solver such as in [201] may help to remove this approximation.

0.19 Parallel Model and Optimizations

0.19.1 Parallel Model

The GTC code uses the Particle-In-Cell method, which has proven to be most suitable for massively parallel computers. Tremendous progress has been made since the days when our gyrokinetic particle was first ported to The Connection Machines.[202] Now, the original circular-geometry version of GTC can run efficiently on various MPP platforms with ten of thousands of processors, due to several levels of parallelism. The original parallel model for GTC was a one-dimensional domain decomposition in the toroidal direction (long way around the torus), allowing a division of grid and particles between several processors. Although this algorithm scales to a large number of processors, it is limited by the number of grid points, or planes, used in the toroidal direction. As mentioned above, our highly-optimized field-line-following mesh dramatically reduces the number of grid points required in the toroidal direction (N_t) since it only needs to adequately resolve the parallel structure of the modes and these modes have very long parallel wavelengths ($k_\parallel \ll k_\perp$). $N_t = 64$ is used for most simulations although we sometimes go up to 128 for some strongly-shaped devices. When GTC was ported to the large IBM SP Power3 system at NERSC several years ago, another level of parallelism was added to take advantage of the computers large shared memory nodes. Loop-level work-splitting was implemented in the code using OpenMP directives. With 16 processors per node, this allowed GTC to scale to 1024 processors (64×16). With the advent of even larger MPP systems, such as the Blue Gene/L and the Cray XT3, yet another level of parallelism was added to GTC

and allowed it to scale to a very large number of processors. These new systems do not currently support thread-level parallelism so we were unable to take advantage of our OpenMP loop-level parallelism. In the new parallel model, particles within each toroidal domain can now be evenly distributed between several MPI processes, making it possible to use a very large number of particles and run very high phase space resolution simulations on tens of thousands of processors. This method scales extremely well. The present version of GTC-S is based on the OpenMP version of the original circular-geometry GTC so it lacks the third level of parallelism that would allow it to run on more than 64 processors on MPI-only platforms such as the Cray XT. GTC-S has also been extensively modified from the original GTC. It now includes numerical equilibrium and noncircular gyroradius, which makes code optimization even more difficult.

0.19.2 Optimizations

As part of the Joule exercise, a team of experts in code optimization from the SciDAC Performance Engineering and Research Institute (PERI) was assign to our project. The main participants from University of Oregon, University of Tennessee-Knoxville, and Rice University proceeded to use profiling tools such as TAU to study the performance characteristics and bottlenecks of the GTC-S code on the Opteron processor, which is the processor on the Cray XT system, and also the Itanium for comparison. Several levels of source modifications were tested, ranging from low to high impact, where high impact refers to code-wide alterations having a significant impact on the overall code development. Modifications such as loop unrolling, loop fusion, and array dimensions reordering lead to a 13% performance improvement on the Opteron processor and up to 33% on the Intel Itanium. Although these changes have yet to be implemented in the production version of GTC-S, we intend to include most of them by considering their performance improvement factor versus their impact on the upcoming upgrade of the Cray XT system to quad-core processors. This new processor will require the vectorization of the main loops in the code to achieve a significant percentage of peak performance on each core. The modifications favoring this aspect will certainly be implemented in the code.

It is worth pointing out that the performance tools used in this optimization work were essential to identify the regions where to focus the effort, to help determine which modification to implement, and also to assess the benefit of each of these modifications. The PERI team carried out a thorough performance study of the GTC-S code that helped the developers understand some subtle aspects of performance improvement on the current processor architecture. They delivered concrete solutions with measured increases in performance.

0.20 Tasks Undertaken to Achieve Joule Milestones

In order to achieve the milestone calling for a doubling in the performance of the GTC-S code, we carried out the following tasks during FY07:

- The IBM-SP version of GTC-S was ported to the Cray XT platform. This required the installation of the SPRNG parallel random generator and the PSPLINE packages on the NCCS Cray XT Jaguar. It also required the replacement of all the calls to the NAG numerical library, which is a proprietary library not installed on Jaguar. With help from ORNL staff (S. Klasky and E. DAzevedo), interfaces to the NAG calls in the code were written in order to replace each call with equivalent routines from the portable CMLIB library. This was done by the end of Q2.

- The MPI-based particle-distribution model was implemented in GTC-S to dramatically increase scaling. We show below how GTC-S achieved a 128$\times$ improvement in concurrency by previously being limited to only 64 cores on Jaguar to now running on 8192 cores. This task was time consuming due to the implementation, debugging, and verification steps. The implementation was done by Q3 and verification by Q4.

- At the newly achieved concurrencies of 8192 cores, I/O performance and data management become a very important issue. An asynchronous parallel I/O method developed by the end-to-end solution group at NCCS/ORNL was implemented in GTC-S for the checkpoint-restart files. The impact of writing these files was significantly decreased and the data management was simplified.

- In-depth performance analysis of GTC-S was carried out and lead to several source code tuning. We had the help of members of the SciDAC Performance Engineering Research Institute (PERI) to achieve this task.

0.20.1 Performance Metric

The metric that we use to measure the performance of our PIC code is the number of particles advanced one step per wall clock time. We call this the compute power and it is evaluated by dividing the number of particles in the simulation volume by the wall clock time per numerical step. This is measured by keeping the computational grid fixed since the size of this grid is entirely determined by the size of the fusion device under study and the physics being investigated. The compute power is evaluated by the following formula:

Compute power = no. of particles $\times$ no. of steps / wall clock time

For a weak scaling study, the amount of computational work per processor, or core, is maintained by increasing the number of particles proportionally to the number of cores. When going from 64 cores to over 10,000 cores, only weak scaling makes sense since a fixed problem size would quickly run out of computational work as the number of cores is increased. For a fixed device size, a larger number of particles is required in order to run the simulation for a longer time while keeping the finite particle fluctuations to a low level.

0.20.2 Problem Statement

The problem statement for our Joule exercise is a microturbulence simulation of ion temperature gradient instability for a specific experimental shot of the DIIID tokamak located in San Diego, CA. DIIID is particularly interesting due to its relevance to the international ITER project. Both devices have very similar cross-section geometry, although ITER will be a much larger tokamak. The DIIID results will be valuable when comparing with ITER once that experiment is up and running. We use DIIID experimental shot 122338 with the temperature and density profiles taken at 1.6 seconds after the beginning of the discharge. Our global simulation consists of 64 toroidal planes with 78,811 grid points on each plane, for a total of a little over 5 million grid points. For our weak scaling study the number of particle per core is kept constant at 10 particles per cell while the grid remains fixed. The 64-core run uses a single core per toroidal plane so the total number of particles per cell is only 10. The 128-core run uses 2 cores per plane, each having 10 particles per cell which adds up to a total of 20 particles per cell. The number of particles thus increases with the number of cores as to maintain the same amount of computational work on each core. The 8192-core run uses 1,280 particles per cell, for a total of 6.4 billion particles in the simulation volume. We run for only 100 time steps, which is sufficient to assess the performance and scalability of the code.

We changed our simulation parameters for Q4. The Q2 test case used 30 particles per cell for the 64-core run instead of 10, and 64 radial grid points on each plane instead of 128. 10 particles per cell per core is a more realistic number for the production simulations that we will carry out at high concurrency. As for the higher radial resolution on each plane, it was determined that the specific DIIID shot being simulated gives better results with the higher grid resolution.

0.20.3 Initial Performance Results at the End of Q2

As a first step towards improving GTC-S, it was ported to the Cray XT system at the National Center for Computational Sciences (NCCS) and initial performance was measured for reference. We ran 100 time steps of an ITG simulation for DIIID tokamak experiment on 64 processors. This device is currently running at General Atomics in San Diego, CA, and has relevance to the ITER project since its cross-section is very similar although much smaller.

The numbers for the initial test run are (for all 64 processors):

- Compute Power = 8.138M particles/(second/step)
- user time = 494.6 sec
- Instructions per cycle = 42.22
- Hardware Floating point operations per cycle = 9.00
- Hardware Floating point operations per user time = 23.41 GFlops/sec
- Percentage of peak =7.0%

No. of cores	user time(sec)	No. of Instr.	No. FP Ops
64	472.1	60.2×10^{12}	13.7×10^{12}
512	476.8	488.5×10^{12}	109.5×10^{12}
4096	490.9	4098.0×10^{12}	876.3×10^{12}

Table 6: Hardware counter measurements of instrumented version of GTC-S using Cray Performance Analysis Tool (CrayPAT) in virtual-node mode (using both cores per node).

- MIPS = 109766.5 M/sec

Time spent in the main subroutines of the code is as follow:

```
CPU TIME USAGE (in SEC):
 pusher    shift     charge    poisson   smooth    field     load      total
 2.457E+02 3.221E+01 1.139E+02 1.570E+01 3.399E+01 1.286E+00 5.101E+01 4.946E+02

 WALL CLOCK TIMES (in SEC):
 pusher    shift     charge    poisson   smooth    field     load      total
 2.457E+02 3.222E+01 1.139E+02 1.570E+01 3.399E+01 1.284E+00 5.101E+01 4.946E+02
MAIN LOOP TIME(SEC): 4.946E+02
```

0.20.4 Performance Results at the End of Q4

Figure 9 shows a weak scaling study of GTC-S that demonstrates the dramatic improvement in concurrency after the implementation of the new level of parallelism. At the beginning of FY07, the GTC-S simulations on the Cray XT system, Jaguar, were limited to 64 processors due to the lack of thread-level parallelism support on this type of MPI-only system. By implementing a new MPI-based particle distribution method, the code can now scale to 8192 cores and beyond. The Compute Power goes from 12.8 million particles/(second/step) for 64 cores to 1.5 billion particles/(sec/step), indicating very good scaling. It is worth noting that the Compute Power on 64 cores is 57% higher than the test case in Q2. This is due in part to the faster XT4 system but also to new optimizations implemented in the code.

We also performed test runs using an instrumented version of the code using the Cray Performance Analysis Tool (Cray PAT) in order to have hardware counter measurements. Table 0.20.4 gives the total number of instructions (No. of Inst.) and floating point operations (FP Ops) for the 3 of the runs of Figure 9. We show that the weak scaling is also demonstrated by the hardware counters as the number of instructions and floating point operations increase proportionally with the number of particles and cores.

0.21 New High Performance Parallel I/O in GTC-S

One of the newest additions to the GTC-S code is the change in the I/O layer, especially for the checkpoint-restart files. At regular intervals, each MPI process in GTC-S opens

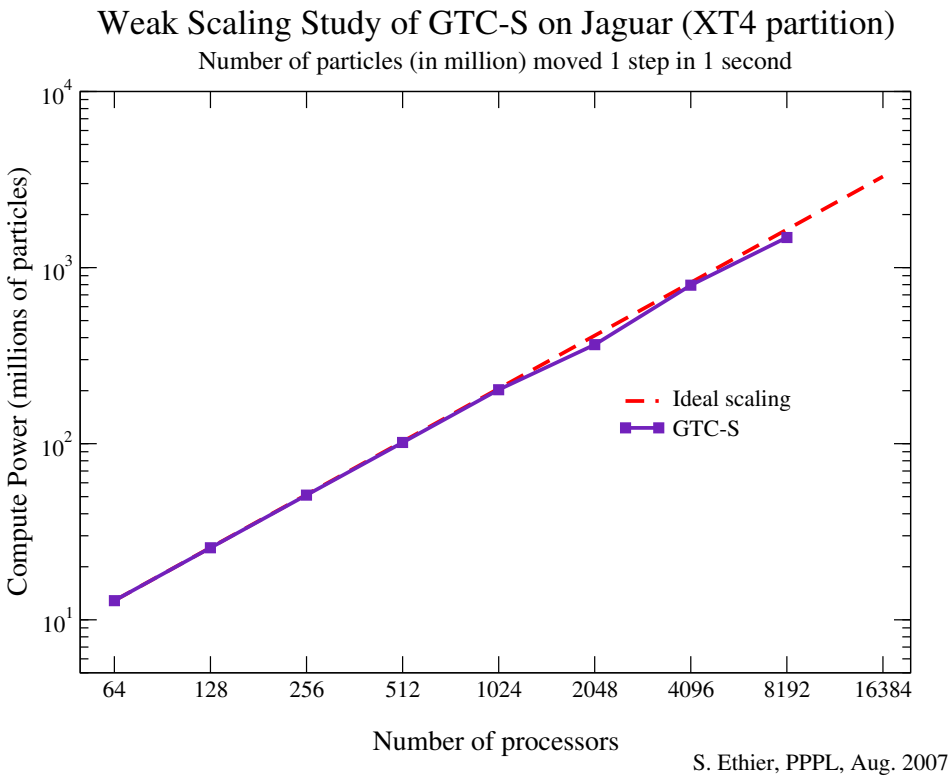


Figure 9: *Weak scaling study of the new GTC-S parallel algorithm. The device size is fixed and the number of particles increases proportionally with the number of cores.*

an individual file and writes in it all the data required to restart the run from this point in the simulation. Using standard unformatted Fortran I/O to write the data, this method worked relatively well for a number of processors less than a thousand or so. For higher concurrencies, the large number of restart files quickly becomes a problem for data management and performance. Simple Fortran I/O, while convenient, can lead to extreme slowdowns at scale. For instance, tests performed by Steve Hodson at Oak Ridge National Laboratories demonstrate [204] that even with as few as about 10,000 cores, there can be substantial performance degradation. For example, for 5 MB writes/reads with 8640 writers, aggregate bandwidths are measured as 1.4 GB/s writing and 2.5 GB/s reading. For 10 MB writes/reads with 8640 writers, bandwidths are measured as 2.3 GB/s writing and 3.1 GB/s reading. Yet, when reducing the number of writers to 960, aggregate performance increases to 10.1 GB/s writing and 10.3 GB/s reading for 5 MB sizes and 7.2 GB/s writes and 9.0 GB/s reads for 10 MB. Another issue is poor usage of parallel IO formats like HDF5 or parallel netCDF, which can result in extreme slowdowns if IO calls are not optimized properly. This is because contention issues with the simultaneous use of the storage systems metadata services can significantly impact I/O performance. Cluster File Systems' own tests have demonstrated this problem to be sufficiently significant to offer distributed metadata services in version 2.x of the Lustre system. The performance issues raised above can be addressed with a variety of techniques, including judicious use of asynchronous I/O techniques, best practices implementations of various APIs, partial serialization of metadata operations, managing the sizes of writes and the number of concurrent writers, and scheduling I/O and communication actions to best exploit the MPPs communication subsystem and I/O capabilities. Yet, to exploit such methods, end users must have detailed knowledge of the underlying machine architecture, I/O and communication system configurations, the communication and I/O behaviors of their parallel codes, and potentially, the runtime behavior of other codes sharing the machine. Better solutions are needed to efficiently operate and exploit the modern, high end MPPs now being deployed and used in U.S. high performance computing centers.

The Asynchronous I/O API (AIO API) designed by our group provides an interface almost as simple as a standard Fortran write, but in contrast to such direct methods, it can be configured to use optimized MPI-IO calls, simple Fortran I/O calls, or asynchronous methods such as the Georgia Tech DataTap servers [205] and the Rutgers DART system [206]. More importantly, internal instrumentation of the API can be used to better schedule MPP programs I/O actions and help deal with potential congestion or slowdown caused by the concurrent actions of multiple MPP applications. The key consideration for the development of the AIO API was that we do not wish to supplant any existing data storage formats or IO mechanisms. Instead, it is intended as a platform for simple, efficient use of the majority of the functionality of these standards without the complexity of learning the intricacies of each different system. Expertise built into this layer effectively and efficiently manages the underlying mechanism selected for each data grouping, using both static and dynamic machine and program knowledge to optimize data transfer operations. The goal is to free the scientist from the intricacies and complexities of I/O tuning. Additional important features of this API include the following: (1) compact, binary data transmission, (2) feedback mechanisms for scheduling storage-related I/O, (3) external configuration of data col-

lection and storage/processing mechanisms, and (4) value-added, additional in-flight and offline/near-line data processing on I/O data.

AIO encodes data in a compact, tagged, binary format for transport. This can either be written directly to storage or parsed for repackaging in another format such as HDF-5 or netCDF. The format consists of a series of size-marked elements, each with a set of tags-values pairs to describe the element and its data. For example, an array is represented by a tag for a name, a tag for a data path for HDF-5 or similar purposes, and a value tag. The value tag contains the number of ranks, the dimensional magnitude of each rank, the datatype, and the block of bytes that represent the data. Additional mechanisms built into the AIO API provide feedback to the storage implementations concerning the best times to perform IO and also to provide progress indicators to gauge how quickly data transmission must be performed to 'keep up with the code.

0.21.1 GTC-S Incorporation

Currently we have only incorporated our AIO APIs into the restart portion of the code. We have seen that we get 11.5 GB/sec on our benchmark run on 128 processors. The code produces only mzetamax files (usually 64) at each timestep. A very nice feature of this io is that all restart files are marked with metadata so that they can be converted into hdf5 or netcdf files. By splitting the number of files into mzetamax, it allows a researcher to do validation with the particles in a particular spatial location, by only reading in 1 restart file. This also permits us to lessen the time it takes to get this information from HPSS. Our I/O routines have been used in the GTC (classic) code, and have been shown by Hodson that we can get 20GB/sec in writing the restarts.

0.21.2 Initial Performance

We ran our initial test on 1024 processors, writing out mzetamax=64 files per time dump. We calculated the I/O rate from timers around the initial open and close calls in our aio APIs. We then showed the stability of our synchronous I/O by showing this for 12 timesteps. Our results are shown in table 0.21.2. We can see that the average I/O rate = 12.98 GB/sec, with standard deviation of approximately 1 GB/sec. The deviation is because we only ran on 1,024 processors while other users were on our system. We are going to look more into optimizing the I/O synchronous method, since we have optimized this at a little over 20GB/sec for a 15K processor GTC run.

0.22 Summary and Future Plans

With GTC-S, we have presented a generalized model which incorporates important realism of tokamak experiments into nonlinear gyrokinetic simulations of plasma turbulence. These include a systematic treatment of plasma rotation and equilibrium $\mathbf{E} \times \mathbf{B}$ flow, realistic plasma profiles and corresponding MHD equilibria. The general geometry simulation capability has been developed with the following favorable features: i) By rescaling the radial coordinate, the grid size in the perpendicular direction is correlated with the local gyroradius which, varying substantially from the core to the edge,

Time Elapsed (s)	File size (GB)	GB/sec
4.37	58.88	13.48
4.98	58.88	11.83
4.06	58.88	14.50
4.38	58.88	13.46
4.85	58.88	12.15
4.90	58.88	12.03
4.97	58.88	11.84
4.24	58.88	13.89
4.24	58.88	13.81
4.43	58.88	13.29
4.72	58.88	12.47

Table 7: I/O rates for the new asynchronous parallel I/O method used in GTC-S to write out the restart data. Only 64 files are created instead of one per MPI process.

defines the spatial scale of turbulence at different locations. ii) Gyrokinetic transformations of potential and charge density between particle and guiding center positions are calculated with a finite ratio (B_θ/B) correction which is a significant geometry effect on the turbulence calculation, particularly for spherical torus devices. iii) The applied equilibrium $\mathbf{E}\times\mathbf{B}$ flow with the spatial scale of the plasma minor radius, which is believed to play an important role in determining the turbulence level, is calculated from our first-principles based particle simulation of global neoclassical dynamics with important finite orbit effects. Working with a symmetry coordinate system, we can construct a relatively regular mesh in real space for strongly shaped toroidal plasmas. This also facilitates straightforward visualization. In the large aspect ratio circular concentric geometry limit, cross benchmarks of the linear and nonlinear characteristics, such as real frequency, growth rate, steady-state heat flux and zonal flow amplitude, of ITG turbulence have been carried out to validate the general geometry model and simulation.

Our immediate task on the physics front is to use GTC-S to study electron temperature gradient (ETG) turbulence for NSTX at PPPL. An experimental campaign is now underway to measure the radial spectrum of ETG turbulence and our simulation in this area will be most timely. The problem has been studied with GTC with circular geometry, for example, Lin et al. [203] and found that the ETG transport is too small to explain the experimental observation from NSTX. With the new GTC-S, we can the chance to study the problem in realistic tokamak geometry. In this exercise, we plan to use over 10,000 nodes on Jaguar with more than 10 billion particles to simulate a substantial slice of the NSTX plasmas. With the improved performance of GTC-S, the simulation will probably take more than several hundred thousand processors hours.

0.23 Acknowledgement

This work was supported by the U.S. Department of Energy (DoE) Scientific Discovery through Advanced Computing (SciDAC) Center for Gyrokinetic Particle Simulation, and by DoE Contract No DE-AC02-CHO-3073.

S3D Description

0.24 S3D : Sandia INCITE Team

0.25 Q2 Problem Information

0.25.1 Background and Motivation

In many practical applications for power generation, such as stationary gas turbines, there has been a strong interest in achieving lean premixed combustion. Land-based stationary gas turbines constitute a significant portion of the electrical power generation industry. The advantages of operating at very lean mixture conditions are high thermal efficiency and low emissions of NOx due to lower flame temperatures. However, combustion at conditions close to the lean flammability limit has the drawbacks of local extinction, emissions of unburned hydrocarbons and large amplitude oscillations in pressure that can result in mechanical damage to the turbo-machinery used in power production. A fundamental understanding of the dynamics of premixed flame propagation will help advance the combustion modeling capability and thereby achieve the engineering design goals associated with lean premixed combustion.

0.25.2 Approach and Goals

Combustion at lean premixed conditions predominantly occurs in the flamelet and thin-reaction zones regimes of premixed combustion [215]. In contrast to the flamelet regime, the thin reaction zones regime is characterized by strong turbulence-chemistry interactions which are not well understood and are very difficult to model. The thin reaction zones regime is also the more appropriate for our numerical method given the overlap in turbulence and flame length scales. We perform Direct Numerical Simulation (DNS) to understand the complex interactions between the turbulence and the flame in the thin-reaction zones regime. The three-dimensional simulations have been performed using a canonical slot-burner Bunsen flame configuration and detailed methane-air chemistry, a good approximation to natural gas, the primary fuel used in gas turbine power production. The simulations were performed for a duration long enough to achieve statistical stationarity, necessary for model development. The ratios of turbulence intensity-to-flame speed and integral length scale-to- flame thickness determine the Reynolds and Karlovitz (flame stretch determining flame area produc-

tion) numbers, important parameters which govern premixed combustion. A parametric study was performed by varying the inflow turbulence parameters to determine the dependence of turbulent flame characteristics on Reynolds number and Karlovitz number. The resulting data will shed light on the role of small eddies penetrating into the preheat layer ahead of the reaction zone and its effect on the flame thickness.

0.25.3 Problem Configuration

The simulation was performed in a slot-burner Bunsen flame configuration. The slot-burner Bunsen configuration is especially interesting due to the presence of mean shear in the flow and is similar in configuration to a burner used in experimental studies, for example by Filatyev [216]. This configuration consists of a central reactant jet through which premixed reactants are supplied. The central jet is surrounded on either side by a heated coflow, whose composition and temperature are those of the complete combustion products of the reactant jet. This arrangement is similar to the pilot flame surrounding slot burners commonly used in experiments [216]. The reactant jet was chosen to be a premixed methane-air jet at 800K and ϕ = 0.7. The unstrained laminar flame properties at these conditions computed using PREMIX [217] are as follows:

1. Flame speed, S_{L} = 1.8 m/s
2. Thermal thickness based on maximum temperature gradient, δ_{L}=0.3mm
3. Full width at half maximum (FWHM) of heat release rate, δ_{H} = 0.14mm, and
4. Flame time scale, τ_{f}=$\delta_{\mathrm{L}}/S_{\mathrm{L}}$=0.17ms.

One of the reasons for choosing a preheated inflow condition is that the cost of computation is inversely proportional to the Mach number at the inflow. Preheating the reactants leads to a higher flame speed and allows a higher inflow velocity without blowing out the flame. Also, many practical devices such as internal combustion engines, gas turbines and recirculating furnaces operate at highly preheated conditions. One important consequence of preheating is that the reaction zone is broadened at 800K ($\delta_{\mathrm{L}}/\delta_{\mathrm{H}}$=2) compared to 300K ($\delta_{\mathrm{L}}/\delta_{\mathrm{H}}$=3). However, the preheat temperature chosen here is low enough that flameless combustion [218] does not occur.

A parametric study was performed to investigate the effect of increasing turbulence intensity on lean premixed combustion. The problem configuration, mixture equivalence ratio and temperature are the same for all three simulations. However, they differ in the domain sizes and inflow turbulence conditions. The simulation parameters are given in Table 8. A uniform grid spacing of 20μm was used in the streamwise, x, and spanwise, z, directions, while an algebraically stretched mesh in the transverse,y, direction was obtained from $y(s) = f(s) \times s$, where s is the equi-spaced computational grid and $0 \leq s \leq 1$. The stretching function is given by,

$$f(s) = \beta s + (1 + \tanh s - s^* \sigma) \left(e^{ks} - \beta s \right), \tag{263}$$

where $k = \ln(\beta s^*)/(s^* - 1)$. The resultant mesh was mirrored across the jet centerline ($y = s = 0$) to obtain a symmetric mesh. The form of the stretching function along

Table 8: Simulation parameters

	Case A	**Case B**	**Case C**
slot width (h)	1.2mm	1.2mm	1.8mm
Domain size in the streamwise, crosswise and spanwise directions	$12h \times 12h \times 3h$	$20h \times 12h \times 3h$	$20h \times 12h \times 3h$
Number of grid points	52 Million	88 Million	195 Million
Turbulent jet velocity ($\bar{U}$)	60m/s	100m/s	100m/s
Laminar coflow velocity	15 m/s	25 m/s	25 m/s
Jet Reynolds number ($\mathrm{Re}_{\mathrm{jet}} = \bar{U}h/\nu$)	840	1400	2100
Turbulence intensity[4] (u'/S_{L})	3	6	10
Turbulence length scale[2,4] ($l_{\mathrm{t}}/\delta_{\mathrm{L}}$)	0.7	1	1.5
Integral length scale[3,4] ($l_{33}/\delta_{\mathrm{L}}$)	2	2	4
Turbulence Reynolds number($\mathrm{Re}_{\mathrm{t}} = u'l_{33}/\nu$)	40	75	250
Karlovitz Number $(\delta_{\mathrm{L}}/l_{\mathrm{k}})^2$	100	100	225

[1] Kinematic viscosity at the inflow conditions, $\nu = 8.5e-5m^2/s$, is used to compute Reynolds number.
[2] Turbulence length scale l_{t} is estimated as $l_{\mathrm{t}} = u'^3/\tilde{\epsilon}$, where $\tilde{\epsilon}$ is the average turbulent kinetic energy dissipation rate.
[3] Integral length scale l_{33} is defined as the integral of the auto-correlation of the spanwise component of velocity in the spanwise direction.
[4] The turbulence scales evolve from the synthetic turbulence specified at the inflow. The u', l_{t} and l_{33} values reported here are at the 1/4th streamwise location along the jet centerline.

with the choice of constants, $\beta = 0.55$, $s^* = 0.75$ and $\sigma = 1/16$, yields a mesh that has a uniform spacing of 20μm in the center of the domain over a region $5h$ in width. Here h denotes the slot width. The increase in grid spacing, $(\Delta_{i+1}/\Delta_i - 1)$, in the outer part of the domain does not exceed 2%. While the uniform grid spacing at the center of the jet ensures numerical fidelity and flexibility in post-processing, the boundaries are pushed farther away to reduce their influence on the flame.

0.25.4 Reduced Chemical Mechanism for Lean Premixed Methane-Air Flames

A reduced chemical mechanism for lean premixed methane-air flames was derived, specifically tailored to minimize temporal stiffness while maintaining accuracy. Details on the reduction methodology and validation of the reduced mechanism are presented below and can also be found in Ref. [219, 220].

GRI-Mech 1.2 [221] is a detailed mechanism optimized for accurate prediction of methane-air combustion under a variety of reaction conditions. It has been widely applied in combustion simulations involving methane oxidation. While the accuracy of the GRI mechanism meets the requirement of the current DNS study, due to the heavy demand in CPU time of the 3D simulation it is necessary to reduce the number of species and reactions in the mechanism before it can be efficiently applied. Furthermore, since an explicit integration scheme is used in the current solver, it is necessary to reduce the stiffness induced by highly reactive radicals with short time scales.

Mechanism reduction typically consists of two stages: (i) skeletal reduction which

eliminates unimportant species and reactions from the detailed mechanism, and (ii) reduction based on quasi steady state (QSS) assumptions which eliminate short time scales as well as the number of species. The method of directed relation graph (DRG) was developed in Ref. [222] as the first step in skeletal reduction to quickly remove a majority of unimportant species and reactions in a detailed mechanism. More time consuming reduction methods such as sensitivity analysis can also be applied to further reduce the skeletal mechanism to the minimum size. The skeletal mechanism obtained is subsequently further reduced with methods based on time scale analysis, and the quasi steady state (QSS) method can be applied to moderate the stiffness and reduce the number of resolved species. In the present study, the method we developed in a previous work [223] is employed to identify the QSS species.

0.25.5 Mechanism Reduction Methodology

To collect a sample space covering typical reaction conditions in the current DNS study, a set of reaction states were first sampled from two representative homogeneous applications: (i) perfectly stirred reactor (PSR) and (ii) auto-ignition. PSR is a typical extinction application covering moderate to high temperature range where a radical pool already exists, and auto-ignition covers a low to moderate temperature range where a radical pool is formed. Both reaction states in PSR and auto-ignition may be important for studying premixed flames which span the low to high temperature range and vastly different species concentrations. The reaction states were sampled under atmospheric pressure, equivalence ratio from 0.6 to 0.9, and initial temperatures from 1000 to 1800K for auto-ignition and 300K for PSR. The method of DRG is then performed on the sampled reaction states, and the worst case relative error induced to the major species by species elimination was calculated for each species [222].

The dependence of the number of species in the skeletal mechanism as a function of the threshold value ϵ is shown in Fig. 10. As observed and discussed in Ref. [222],

Figure 10: Number of species in skeletal mechanism as a function of the threshold value ϵ in skeletal reduction with directed relation graph. Figure from Ref. [220].

while species with small values of ϵ, say ϵ <0.2, can be safely removed by DRG, elimination of species with larger ϵ might result in significant loss of accuracy due to the possible existence of species group equilibrated by multiple fast reactions. On the other hand, species with ϵ of the order of unity, say ϵ >0.5, are strongly coupled with the major species and should definitely be retained. Therefore, species in the detailed mechanism are divided into three groups in DRG reduction as shown in Fig. 1: The first group consisting of 11 species, HCCO, CH_2CO, HCCOH, C_2H, C_2H_2, CH_3OH, C_2H_3, C, C_2H_4, C_2H_5, and Ar, with ϵ <0.2, is unimportant and can be safely removed; the second group with ϵ >0.5 is important species and has to be retained, while the third group with ϵ between 0.2 and 0.5 might be unimportant but is not safe to be removed in DRG reduction.

Additional information, such as the sensitivity of global parameters, for example the flame speed, is required to scrutinize the importance of the species that fall into the

'uncertain group' of DRG, and such additional analysis is typically significantly more time consuming than DRG. While in many situations only very few species in the uncertain group can be further removed after DRG, the heavy CPU time demand by the current 3D DNS justifies the necessity to eliminate every non-crucial species in spite of the effort required by this additional skeletal reduction stage. In the current work, considering that there are only seven species, namely H_2O_2, C_2H_6, CH_2OH, CH_3O, CH, $CH_2(S)$, and CH_2, in the uncertain group, a brute force but effective method, by comparing the global sensitivity of flame speed with respect to the elimination of each of the seven species, was applied to reduce the skeletal mechanism to its minimum size, and it was found that H_2O_2, C_2H_6, CH_3O, and CH, can be further eliminated while maintaining the accuracy of the flame speed. The final skeletal mechanism therefore contains 17 species and 73 elementary reactions, which is further reduced by QSS assumptions with the CSP method developed in Ref. [222]. Four radicals, CH_2, $CH_2(S)$, HCO, CH_2OH, were identified as global QSS, resulting in a 9-step reduced mechanism. Since the algebraic equations of the QSS relations are rather compact, the QSS species concentrations were solved analytically without the need for iterations. As a result, the overall convergence was obtained at a lower computational cost. Furthermore, the lack of iterations ensured good performance on vector computing platforms.

0.25.6 Mechanism Validation

Since some crucial species for ignition processes, such as H_2O_2 and CH_3O, are not retained in the reduction, the reduced mechanism is not expected to be valid for ignition applications or flames with re-ignition, which is not an aspect of the current DNS study. Therefore the 9-step reduced mechanism was validated only against the detailed mechanisms for PSR and laminar premixed flames. Figure 11 compares the temperature profiles in PSR for equivalence ratios of 0.6 and 0.9 between the 9-step reduced mechanism and GRI 1.2. The good agreement indicates that the crucial species and

Figure 11: Comparison of temperature profiles as a function of residence time in perfectly stirred reactors for lean CH_4-air under STP, calculated with detailed and reduced mechanisms. Figure from Ref. [219].

reaction pathways for high temperature chemistry are intact in the reduced mechanism. The validation of the laminar flame speed was then compared in Fig. 12. It can be

Figure 12: Comparison of laminar flame speed as a function of equivalence ratio for lean methane-air under STP, calculated with detailed and 9-step reduced mechanisms. Figure from Ref. [219].

seen that while the error gradually increases as equivalence ratio approaches unity, the agreement is very good near ϕ=0.7, which is the target mixture composition for the current DNS study.

0.25.7 Numerical Method

The simulations were performed using the DNS code S3D, which has been documented as part of the FY05 Joule exercise. A brief description of the code is given below. Please see section[0.27.2] for details.

S3D solves the fully compressible Navier Stokes, species and energy equations with a fourth-order Runge-Kutta method for time integration and an eighth-order explicit spatial differencing scheme [224, 225]. A tenth-order filter was used periodically to damp any spurious high-wave number oscillations. The mixture specific heat is determined locally as a function of mixture composition; that is, $C_p = \sum_k C_{p,k} Y_k$, where each $C_{p,k}$ is curve-fitted as a function of temperature using the Chemkin thermodynamic database [226]. The molecular viscosity is also temperature dependent and constant Lewis numbers for individual species are used.

The flame is anchored at the inflow plane by specifying the species mass fractions and temperature from an unstrained laminar flame solution using a progress variable lookup. A hyperbolic tangent function was used to obtain a smooth variation of progress variable between the unburned and burned conditions. A turbulent velocity field was synthesized by specifying the length scale, magnitude of velocity fluctuations and spectral energy density. The resultant velocity field was added to the mean inflow velocity profile and used as the velocity inflow boundary condition based on Taylor's hypothesis.

Navier-Stokes characteristic boundary conditions (NSCBC) [227, 228, 229, 230] were used to prescribe the boundary conditions. The boundary conditions were periodic in the spanwise direction (z), non-reflecting inflow and outflow in the streamwise direction (x), and non-reflecting outflow [227] in the transverse direction (y). Based on the jet inlet velocity and the streamwise domain length, a flow-through time is 0.24ms. The solution was advanced at a constant 2ns time-step for three flow through times for case A and two flow through times for cases B and C. The first flow through time was neglected to account for initial transients when performing data analysis. Data from 61 equally spaced time instants from the remainder of the simulation was used to obtain the statistical results presented in the next section. Averaging is performed in the homogeneous direction (z) and time. Symmetry across the centerline is exploited where feasible.

0.25.8 Q2 Benchmark Run Description

As part of the Joule exercise, the above mentioned simulation was benchmarked in Q2 of FY2007. In particular, the simulation case C (see Table 8) was benchmarked on the Cray XT4 system at NCCS, 'Jaguar'. The simulation case C had 195 million grid points and was performed on 7200 cores of the system. The simulation was advanced through 3000 timesteps and performed I/O 4 times during the benchmark simulation. The code was also instrumented with the CrayPat tools which provide the hardware performance counters through the pat_hwpc utility. As described in the FY2005 S3D Joule report, a useful measure of S3D performance is the 'cost of simulation per grid-point per time-step' for a given problem configuration. The Q2 benchmark run required 5570 seconds on 7200 cores. Therefore, cost of simulation = 5570 * 72003600$\times 195 \times 10^6 \times 3000$

$= 1.9 \times 10^{-8} \mathrm{core - hours/grid - point/time - step}$

The pat_hwpc utility reported the floating point computation rate to be 12.4% of the peak performance.

The current version of S3D code solves for the variables on a 3D Eulerian grid, and only this data is output during I/O. The following section discusses a science result obtained from analyzing the data produced from the simulation at the current capability. It further discusses the additional insight that can be obtained from the new capability proposed to be implemented during Q3 and Q4 of FY2007.

0.25.9 Effect on flame thickness

It has been proposed [231] that in the TRZ regime, the turbulent eddies can penetrate the preheat zone and increase the mixing process, leading to a thicker flame. However, the results of experimental studies in this regime are contradictory, with some reports of thicker flames [232, 233] and others of thinner flames [234, 235]. See Ref. [236] for a detailed discussion of the contrasting experimental results on this topic. Computational studies have not yet yielded a definitive conclusion on the flame thickness issue either. Although some computational studies [237, 238] have found that, on average, the flame gets thinner, the results are not conclusive due to the lack of realism in 2D turbulence in Ref. [237] and statistical stationarity in Ref. [238]. Here, we analyze the current simulation results available on the 3D eulerian grid to determine if, on average, the flame thickness increases or decreases relative to a laminar flame.

A reaction progress variable, c is defined based on the mass fraction of O_2. While c is usually defined based on the deficient reactant, in this case CH_4, such a definition here will omit a significiant portion of the oxidation layer, since the heat release is only 66% complete where CH_4 is completely consumed. Therefore, c is defined using O_2 mass fraction. Based on the laminar flame solution at the chosen reactant conditions, the heat release is a maximum at c = 0.65. Therefore, the iso-surface corresponding to c = 0.65 is taken as the flame surface.

The reciprocal of the magnitude of progress variable gradient, $1/|\nabla c|$ yields a flame thickness analogous to the definition used for the laminar thermal thickness δ_L. $|\nabla c|$ is averaged over intervals of c and compared with the unstrained laminar flame profile in Fig. 13. In ref. [220], the conditional mean of $|\nabla c|$ was presented for case A and the results showed that the mean $|\nabla c|$ was lower in the turbulent flame than in a laminar flame, which indicated flame thickening. Here, the same analysis is also applied to cases B and C, to verify if the flame continues to get thicker as the turbulence intensity is increased. A comparison of case A with case B in Fig. 13 shows that the mean gradients are further reduced. This again indicates an increase in flame thickening due to the increase in turbulence intensity from u'/S_L = 3 to 6. However, a comparison of case B with case C shows that there is negligible increase in flame thickness even though the turbulence intensity was increased from u'/S_L = 6 to 10. This is a very interesting result and the reason for the lack of flame thickening at the highest turbulence intensity needs to be ascertained.

To further understand the role of flame surface curvature in thickening of the flame, a balance equation for the iso-surface following derivative of $|\nabla c|$ on the progress

Figure 13: Conditional means of $|\nabla c|$ compared to the laminar flame profile for the three simulations. The means are computed at three chosen streamwise locations corresponding to 1/4, 1/2 and 3/4th of the domain length in the stream-wise direction. Figure from Ref. [239].

Figure 14: Budget of the source terms in Eq. 264 as a function of curvature at the three chosen streamwise locations for Case A. The terms are non-dimensionalized using τ_{f}. Figure from Ref. [220].

variable iso-surface is written as [240],

$$DDt|\nabla c| = -\left(a_{\mathrm{n}} + S_{\mathrm{d}} n\right)|\nabla c| \tag{264}$$

where, the normal strain, a_{n}, is given in terms of the dilatation (a_{d}) and the tangential strain rate (a_{t}) as $a_{\mathrm{n}} = a_{\mathrm{d}} - a_{\mathrm{t}}$. Furthermore, the propagation speed S_{d} is decomposed into its reaction, normal diffusion and curvature components as $S_{\mathrm{d}} = S_{\mathrm{d,r}} + S_{\mathrm{d,n}} + S_{\mathrm{d,c}}$ [241]. From Eq. 264 it is seen that a positive (negative) valued term on the RHS can act as a source (sink) for $|\nabla c|$, thereby leading to thinner (thicker) flame.

A budget of the different terms conditional on curvature at c=0.5 is shown in Fig. 14 for Case A leading to the following observations. First, the dilatation is found to decrease with curvature, due to the negative correlation of S_{d} with curvature [241]. Second, tangential strain rate is also found to be negatively correlated with the magnitude of curvature and therefore highly curved regions have lower values of a_t and vice versa (Note that Fig. 14 shows the *negative* of a_{t}). Furthermore, it is seen that at large positive values of curvature, the strain rate can become compressive. Curved flames cause an induced strain rate due to flow field divergence, which is dependent on the sign of curvature (convex or concave). This contributes to the asymmetry of a_{t} dependence on curvature and the compressive strain found at high positive curvatures and absent at high negative curvatures. Positive values of the tangential strain rate act as a source of $|\nabla c|$ and hence cause thinning of the flame. Third, as the tangential strain rate decays in the downstream direction, the sum of the reaction and normal diffusion components, otherwise known as kinematic restoration term [231], $\partial S_{\mathrm{d,r+n}}/\partial n$, also decays to restore the balance. However, by the '3/4'th position, the kinematic restoration is significantly reduced such that the strain rate is mainly counteracted by dilatation. Fourth, for small curvatures, $-2 < |\nabla \cdot n|\delta_{\mathrm{L}} < 2$, $\partial S_{\mathrm{d,c}}/\partial n$ is relatively small in comparison to the other terms and the tangential strain is primarily counteracted by dilatation and kinematic restoration. However, for large values of curvature, the curvature term is significant and acts as a sink for $|\nabla c|$ leading to thicker flames. Thus, the source of the variation in flame thickness can be explained using Eq. 264. Further insight can be obtained by tracking the history of iso-surface elements as they propagate and convect across the domain [242].

0.25.10 Tracer particle

Tracer particle methods are used commonly in the fluid-mechanics and combustion communities for fundamental understanding of flow and flame physics [243, 242]. Tracer particles are massless particles that are advected in the computational domain concurrent with the solution evolution. In the case of S3D, the equations for the field variables will continue to be solved and advanced in time on the Eulerian grid. The particles will be used to sample and report the relevant data without influencing the main solver. As an example, the tracer particles can be advected at the local Eulerian velocity. Then the instantaneous location of the tracer particle will have to be obtained from time integration of the Eulerian velocity. This is known as Lagrangian fluid element tracking. In a reacting flow, the tracer particles can be advected at the same velocity as a scalar iso-surface, corresponding to either a reacting scalar (such as species concentration or temperature) or a conserved scalar (such as mixture fraction). This is referred to as scalar iso-surface element tracking.

In turbulent combustion the flame is influenced by time-varying flow fields and responds to the flow. For example, in the case of a non-premixed flame, the flame may extinguish or reignite depending on the local scalar dissipation rate. In a premixed flame the fluid dynamical strain tends to increase the flame surface area, while the action of vortices cause wrinkling of the flame surface. Both of these effects change the burning rate per unit area of a premixed flame surface. However, due to the finite-rate of the underlying chemistry, a flame has a finite response time. In other words, it

does not respond to the imposed flow conditions infinitely fast. The flame response has been known to depend on the ratio of the time-scale of imposed flow fluctuations to the flame time scale [244]. Therefore, an instantaneous state of a flame at any location in the domain is determined not only by the local conditions, but also by the history of the conditions that it underwent during its advection through the domain. In order to unambiguously match the cause and effect, it is necessary to know the history of the flame element. To obtain such a history from the data on an Eulerian mesh alone would require a volume of I/O several orders of magnitude higher than the current I/O rate. This would degrade the performance of S3D to an unacceptable level.

Alternatively, the location of several flame elements can be tracked using a tracer particle advection algorithm and the quantities of interest can be interpolated to this location and periodically output to the disk. This approach would involve an increase in computational cost due to the overhead involved in determining the instantaneous flame-element location and the I/O required to output the additional particle data. However, it is expected that it would provide a level of scientific understanding which was not obtainable from the earlier capability.

0.25.11 Flame element tracking

A flame element can be tracked by following the motion of a scalar iso-surface element. In the case of premixed flames, a reactive scalar known as the progress variable defines the flame surface. In the case of non-premixed flames, a passive scalar known as the mixture fraction defines the flame surface. Let ϕ be the scalar and $\phi = \phi^*$ be the scalar iso-surface being tracked. Let $\vec{x}_f$ denote the instantaneous location of a flame surface element. $\vec{x}_f$ can be determined by integrating the equation,

$$d\vec{x}_f dt = \vec{u} + S_d \vec{n} \tag{265}$$

where, $\vec{u}$ is the local Eulerian fluid velocity. S_d is the displacement speed of the iso-surface ($\phi = \phi^*$) given by [240],

$$S_d = D\phi/Dt|\nabla\phi| \tag{266}$$

D/Dt is a material derivative and it consists of the diffusion and reaction terms that are also present in the original set of scalar equations solve by the main solver. $\vec{n}$ is the unit normal vector to the ϕ iso-surface.

The tracer particle algorithm will perform the following steps:

1. Distribute tracer particles on the flame surface at the start of the simulation.
2. Perform accurate time integration of equation 265 to determine the instantaneous particle locations.
3. Retire particles that leave the domain.
4. Retire particles that are in a singular region. (Regions of the domain where $\nabla\phi = 0$ represent singular regions. In these regions, an iso-surface element becomes degenerate. This is also seen from equation 266 where $\nabla\phi$ appears on the denominator. One common source of occurrence for this event is flame merging and pinch-off or other topological changes.)

5. Maintain a constant inflow of particles at the inflow boundary.
6. Periodically interpolate key derived variables to the particle location and output the data.
7. In addition, a parallel implementation will require an exchange of the particles across MPI threads when they move across one MPI domain into another.

After a good implementation of the tracer particle algorithm has been obtained, the following steps will be necessary before the new capability can be used in a production simulation.

1. Validate the accuracy of the tracer particle algorithm by testing it in a simulation of a known flow problem such as the Taylor-Green vortex.
2. Perform simulations with different tracer particle populations to determine the necessary characteristics of the tracer particle population that can provide statistically convergent and acceptable results.
3. Determine the key derived physical quantities that must be interpolated to the particle locations as part of data output.

0.25.12 Data Analysis

Currently S3D writes data using unformatted fortran binary I/O statements. The data is usually archived in HPSS as the simulation progresses. The data is also migrated to a smaller resource such as the Linux cluster at the Combustion Research Facility (SNL) or the SGI Altix system at the NCCS. Since FY2006, the S3D Sandia INCITE team has generated over 40 TB of data from several simulations. Data cannot be held on the scratch space of the analysis systems longer than a couple of weeks. However, the analysis is an iterative process and requires revisiting the data periodically to obtain new information. Everytime the data needs to be analyzed, data is downloaded from the archive and migrated to the appropriate resources. The analysis is then performed using S3D itself. S3D has a built-in data reader for the unformatted binary data which does a reverse of the write procedure. The large 3D data is topologically and statistically aggregated into a condensed form suitable to be viewed as line plots or contour plots. This is then output in small text files suitable for plotting software such as tecplot. The particle tracking module will require an additional data reader for reading the particle data. It will also require additional analysis routines to produce output such as the average time history of flame elements, scalar-velocity correlations, history effects, etc.

0.26 Q4 Problem Information

0.26.1 Ethylene chemical mechanism

To study effects of turbulence-chemistry interactions in turbulent combustion regimes of extreme flame extinction, followed by reignition, confidence in the chemical mechanism is important. A reduced ethylene mechanism was developed from a validated

detailed mechanism [248] by first creating a skeletal mechanism with fewer species using the directed relation graph method and sensitivity analysis. The mechanism was further reduced by defining quasi-steady state (QSS) species using a computational singular perturbation analysis. The resulting mechanism consists of the following 19 transported chemical species: H_2, H, O, O_2, OH, H_2O, HO_2, H_2O_2, CH_3, CH_4, CO, CO_2, CH_2O, C_2H_2, C_2H_4, C_2H_6, CH_2CO, C_2H_6, and N_2.

The reduction strategy was tailored for use in multi-dimensional, turbulent, reacting flow simulation. Specifically, the reduction in the number of transported species, along with minimization of temporal and spatial (e.g. required grid resolution) stiffness, through QSS species, results in significant computational cost savings that makes DNS with complex chemistry feasible. In addition, the QSS approximation is implemented without the need for nonlinear iterative solution, which reduces computational costs and improves numerical stability.

The reduced mechanism was extensively validated for ignition delay time, PSR extinction residence time, laminar premixed flame speed, and species profiles in premixed laminar flames, and opposed jet diffusion flames. Excellent agreement between the reduced and detailed mechanisms were obtained; details can be found in [249].

0.26.2 Configuration and Initial Conditions

The flow configuration is a temporally-evolving, planar slot-jet consisting of a central slab of fuel, surrounded by oxidizer. The configuration was chosen to be similar to the medium Reynolds number case by Hawkes et al. [250] in simulating extinction and reignition of a CO/H_2 jet, allowing for a comparison of the differences between fuel types under similar flow and chemical parameters. Hawkes et al. analysed reactive surfaces of a reactive scalar and found the dominant mode of flame reigntion to occur via edge flame propagation and turbulent flame folding [251]. Edge flame propagation dominated in regions of lower scalar dissipation rate whereas the highest rates of reignition were found to occur by flame folding at higher scalar dissipation rates. Extinction and ignition behavior of ethylene are expected to be significantly different than that of CO/H_2. Hawkes et al. observed monomodal PDFs of reactive scalars (e.g. OH) during extinction and reignition, whereas we have observed bimodal behavior of PDFs of reactive scalars for the ethylene flame during extinction. These results are consistent with the experimental studies of Masri and Bilger, who considered nonpremixed flame extinction with H_2, CO/H_2, and CH_4 flames [252]. CO/H_2 has a broader reaction zone width in mixture fraction, and a lower activation energy than ethylene. Hence CO/H_2 is a more robust fuel than ethylene in terms of its extinction and reignition characteristics, and important differences in the flame structure of the resulting reignition processes may occur.

Table 9 gives the geometric parameters used in the present simulation. In the table, H is the initial jet velocity height, ΔU is the velocity difference between the fuel core and the surrounding oxidizer, and Re_{jet} is based on H, ΔU, and the kinematic viscosity of the fuel core. L and Δx are the domain length and grid spacing, respectively. The jet time τ_{jet} is $H/\Delta U$. Here, x, y, and z denote the streamwise, cross-stream (mean shear), and spanwise directions, respectively. The corresponding boundary conditions are periodic, non-reflecting outflow [253], and periodic, respectively.

Table 9: Temporal ethylene jet simulation parameters.

H (mm)	0.96	L_x/H	12	$u'/\Delta U$ (init)	5%
ΔU (m/s)	196	L_y/H	19	H/L_{11} (init)	3
Re_{jet}	5120	L_z/H	8	τ_{jet} (ms)	0.0049
H_ξ (mm)	1.5%	Δx (μm)	17	τ_{run}/τ_{jet}	140
δ_u (mm)	0.19	δ_ξ (mm)	0.74	Mean timestep (ns)	5

The jet is initialized with a mean shear velocity gradient. To trip the turbulent shear layer, an isotropic, turbulent velocity spectrum (Passot Poquet [254]) was set up in the fuel core region. The spectrum parameters are u' and L_{11} given in Table 9.

The composition field was initialized by specifying a mixture fraction profile between the fuel and oxidizer streams. A hyperbolic tangent transition with a characteristic width δ_ξ was used between the streams. The width corresponds to about 50% of the extinction value. A one-dimensional, strained flamelet solution was solved in the mixture fraction coordinate, with unity Lewis numbers, and the composition, temperature, mixture fraction dependence used to initialize a burning flame in the jet. The composition- and temperature-dependent scalar dissipation rate profile in the one dimensional solution was computed to match the corresponding mixture fraction-spatial profile of the temporal jet [249].

It was desired to match as many parameters between the CO/H_2 simulation and the present simulation with the ethylene fuel. The parameters of interest are the stoichiometric mixture fraction, the density ratio of reactants and products, the kinematic viscosity, and the extinction scalar dissipation. Table 10 shows a comparison of parameters between the fuels. Three values of the stoichiometric mixture fraction of the ethylene fuel are listed. The extinction scalar dissipation rate is the same in each case. To achieve a composition that gave the desired parameters, the total nitrogen in the system was varied. At a given value of nitrogen the stoichiometric mixture fraction was varied by moving nitrogen between the fuel and oxidizer streams [255]. Note the remarkable agreement between the parameters for $\xi = 0.422$ between the CO/H_2 and C_2H_4 cases. In the table, the density ratio is between stoichiometric reactants and products. The number of moles can be normalized in each stream to give the corresponding mole fractions.

Three dimensional test cases were run at low resolution using ξ_{st} values of 0.422 and 0.25. In each of these cases global extinction of the flame occurred. The stoichiometric mixture fraction was reduced to move the flame further away from the shear layer. The present case was run with $\xi_{st} = 0.17$.

0.26.3 Results and Discussion

Figure 15 shows the temperature contours on a span-wise cross-section of the 3D domain at different instances in time. At the initial condition, the flame was fully burning. At later times portions of the flame were extinguished as seen from the reduction in the

Table 10: Comparison of fuel parameters.

	CO/H_2	C_2H_4		
ξ_{st}	0.422	0.422	0.25	0.17
T_o (K)	500	550	550	550
ρ_r/ρ_p	4.2	4.3	4.5	4.8
ν (cm^2/s)	0.416	0.421	0.4	0.37
χ_q (1/s)	2380	2380	2380	2380
T_q (K)	1296	1700	1792	1894
T_{ad} (K)	2376	2345	2453	2569
$n_{O2,\,\xi=0}$	0.6	3	3	3
$n_{N2,\,\xi=0}$	1.8	6.988	8.576	8.417
$n_{N2,\,\xi=1}$	0.8	6.602	3.0	1.369
$n_{C2H4\,or\,CO,\,\xi=1}$	1	1	1	1
$n_{H2,\,\xi=1}$	0.2	-	-	-

high temperature area. Towards the end of the simulation, after 0.48 ms, the extinguished regions reignite again as evident from the marked increase in the area of high temperature regions. Therefore, the simulation achieved its goal of enacting an extinction and reignition event in turbulent non-premixed flames. The data will be analyzed to obtain insights into the mechanisms that govern and contribute to the reignition. Specifically, the role of the scalar dissipation rate due to turbulent fluctuations in the flow field will be studied using statistical measures such as the means and variances of the scalar dissipation rate and temperature, shown in figure 16.

S3D makes use of the increased computational capability in a weak scaling mode by solving a larger problem within a comparable wall-clock time. In particular, a faster computer that can perform a higher number of floating point operations per second is used by S3D to achieve one of the following. (i) An increased grid size which will allow the simulation of higher Reynolds number regimes by capturing a wider range of length scales. (ii) Larger chemical complexity which will allow the simulation of more complex and realistic hydrocarbon fuels. (iii) Longer integration time which allows a more complete temporal development of the solution and a larger statistics sample set for data analysis. During the FY'07 we have performed a larger simulation using a larger computer resource. The characteristics of the current (Q4) ethylene-air non-premixed flame simulation are compared against the previous (Q2) simulation of a methane-air premixed flame in table 11. It is seen that roughly twice the degrees of freedom were simulated in Q4 but within the same wall-clock time by utilizing twice the number of processors than before. Thereby, the S3D team has achieved the Joule metric for the current year.

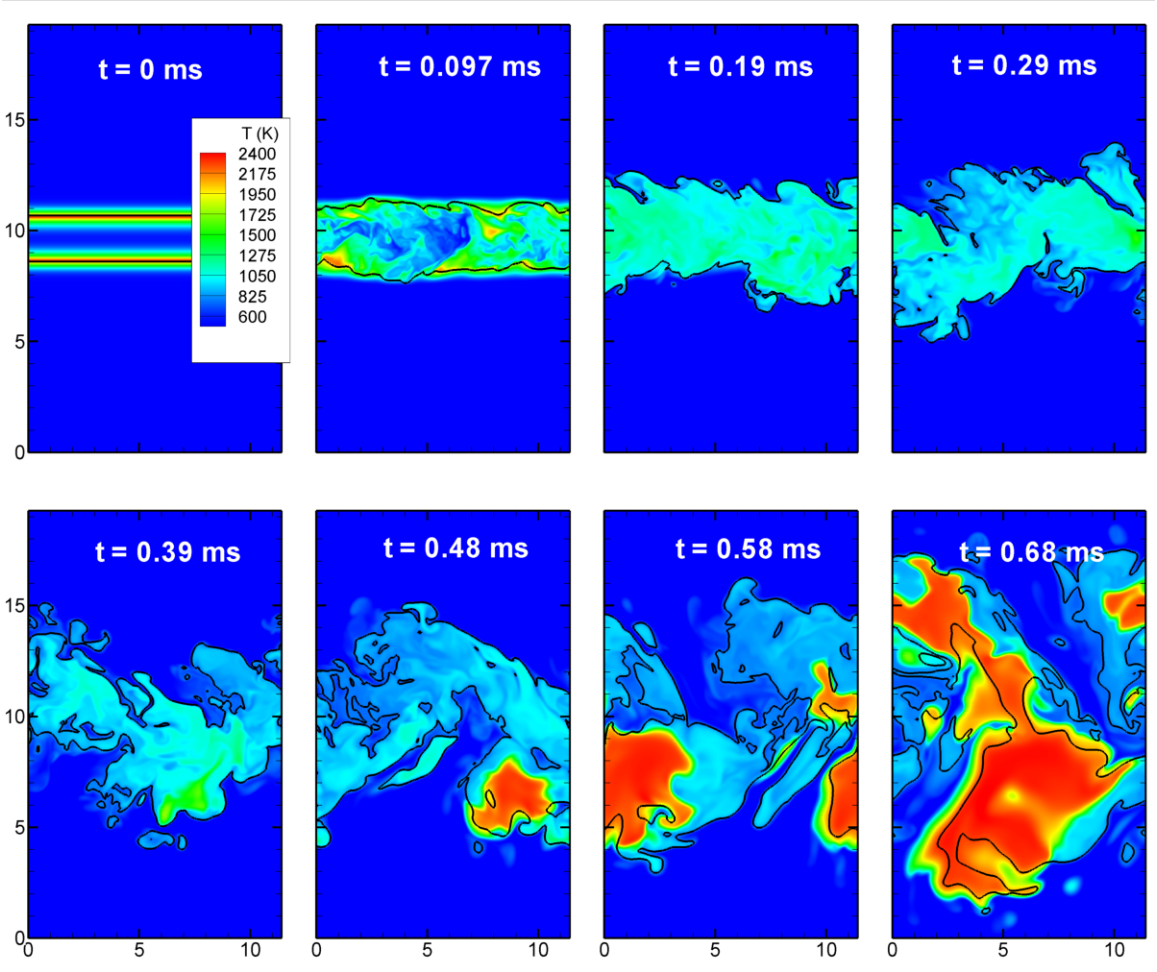
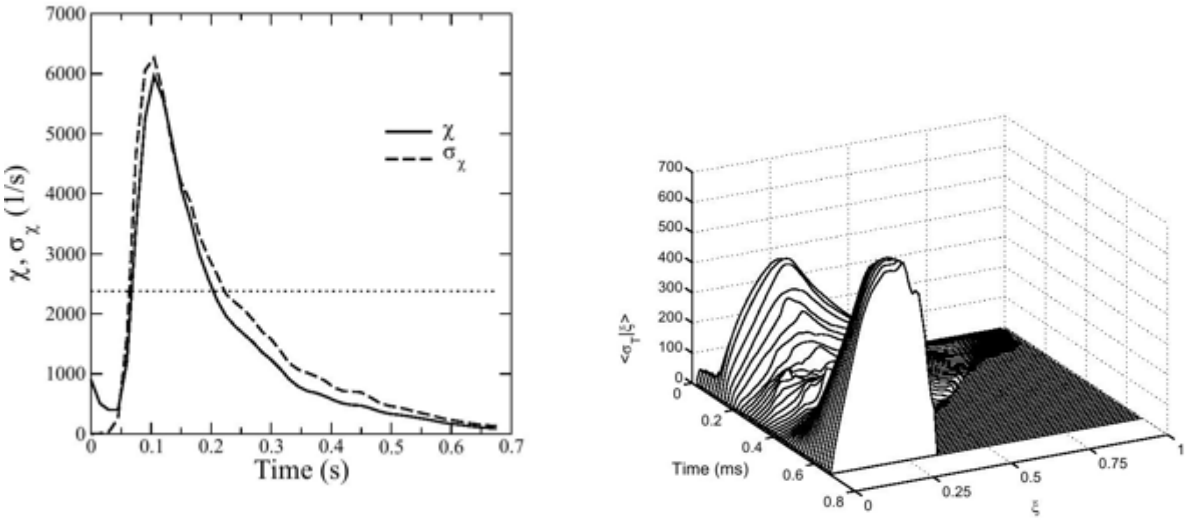


Figure 15: Contour plots of temperature shown in the progression in the levels of extinction and subsequent reignition. Vigorously burning high temperature regions are colored red.



(a) (b)

Figure 16: Conditional statistics on the stoichiometric mixture fraction surface. (a) Mean and standard deviation of scalar dissipation rate. (b) Standard deviation of temperature.

Table 11: Comparison of the Q2 and Q4 simulations demonstrating the factor of two increase in simulation scale.

	Q2	Q4
Chemistry	Methane-air	Ethylene-air
Number of chemical species	13	19
Degrees of freedom per grid point	17	23
Simulation mesh size	195 million	341 million
Total degrees of freedom	3.3 billion	7.8 billion
Number of processors used	7,200	14,112
Processor hours	1 million hours	1.6 million hours
Wall clock days	6 days	5 days
Sustained FLOP/s	4.6 TF	10.2 TF
Fraction of peak performance	12%	14%

0.27 S3D Performance

0.27.1 Introduction

During Q2 of FY'07 a fully resolved direct numerical simulation of turbulent premixed flame was characterized and formed the starting point for this year's joule activity. At the end of FY'07 we have performed a 350 Million grid point simulation of extinction and reignition in ethylene-air non-premixed flames. During this period, the Cray XT system at the NCCS has roughly doubled in its capability while the S3D team has also scaled up the simulation to time-advance more than twice the degrees of freedom than before. The combination of these two advances has enabled the achievement of a new, unique and larger science simulation within similar wall-clock times. During this period, due to the efforts of the SciDAC PERI team, the computational performance of S3D has also been improved. The following report describes the computational improvements to the code, followed by a description of the new simulation and lastly demonstrates the factor of 2 increase in simulation capability achieved during this period.

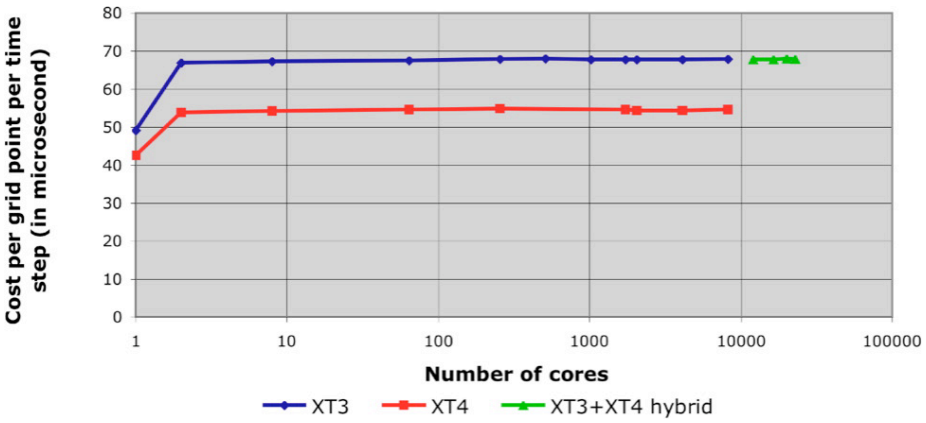


Figure 17: Weak scaling performance of S3D on a Cray XT3+XT4 hybrid at NCCS.

S3D uses almost no collective communication; almost all communication is nearest neighbor point-to-point non-blocking operations with the opportunity to overlap message communication. Thus, S3D exhibits excellent weak scaling performance.[12] Figure 17 shows results from running a model problem that uses a $50 \times 50 \times 50$ grid per core on Jaguar. Runs exclusively on XT4 nodes consistently take approximately 55 microseconds per grid point per time step as the computation is scaled from two to 8192 cores, as shown by the red curve in the figure. While runs exclusively on XT3 nodes yield similar flat weak scaling, the runs take slightly longer – approximately 68 microseconds per grid point per time step – due to the lower memory bandwidth of the XT3 nodes. The blue curve in Figure 17 shows the performance of the XT3 nodes.

[12] Weak scaling is when the problem size is directly proportional to the number of processors.

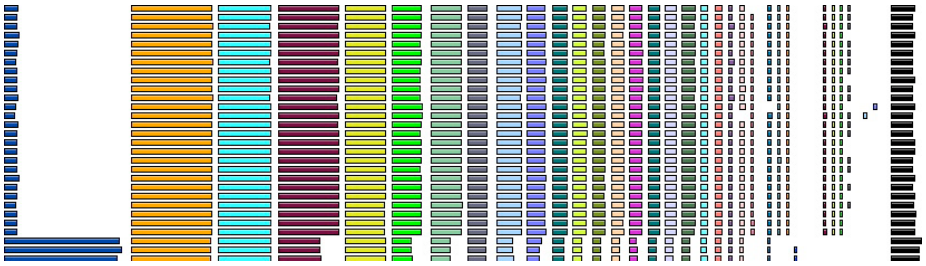


Figure 18: Distribution of exclusive time among S3D's procedures and loops for selected processes in a 6400 core execution.

With Jaguar's current configuration, runs on more than 8192 cores must use a combination of XT4 and XT3 nodes. S3D also exhibits outstanding weak scaling performance on such heterogeneous configurations, as shown by the green curve in Figure 17. However, performance is dominated by the memory bandwidth limitations of the XT3 nodes when a constant problem size per core is used. Thus, the cost per grid point per time step from 12000 to 22800 cores is approximately 68 microseconds, matching the computation rate on the XT3 cores alone.

We performed a detailed performance analysis of runs on heterogeneous allocations using TAU [245]. Figure 18 shows a quantitative breakdown of execution time among loops and procedures for a subset of the processes in a 6400 core execution. Each row of colored bars represents the breakdown of time among activities by a process. Each column of colored bars represents an activity, e.g. time spent in a particular procedure or loop. The processes shown exhibit two general equivalence classes of performance. The processes represented by the lowermost three rows spend substantially longer in `MPI_Wait`, which is represented by the leftmost column of bars. Analysis of the metadata associated with the execution confirmed that these processes were mapped to XT4 nodes. The other processes shown, which were mapped onto XT3 nodes, spend much less time in `MPI_Wait`. Close examination of the figure shows that some columns of bars, such as the second from the left, are nearly identical in length, while for others, such as the fourth, sixth and seventh columns from the left, the bars for the processes on XT3 nodes are noticeably longer. Further study of these data showed that CPU-bound computations take approximately the same time on both XT3 and XT4 nodes, while memory-intensive loops take longer on the XT3 nodes. Overall, these results show that overall application performance on the hybrid system is limited by the memory bandwidth of the XT3 nodes. The slower performance of the XT3 nodes on memory-intensive loop nests causes the XT3 nodes to arrive late at communication events, which is reflected by the longer waiting time on XT4 nodes.

As part of the Joule metric, S3D will run a large problem on the full Jaguar installation. The overall goal is to run a very large problem that demonstrates the addition of new chemistry capabilities to S3D. However, completion of that problem will require outstanding single node and scaling performance. Although S3D already performs very well on the combined machine, the clear performance limitation of the XT3 nodes on overall performance suggest a straighforward mechanism to improve overall perfor-

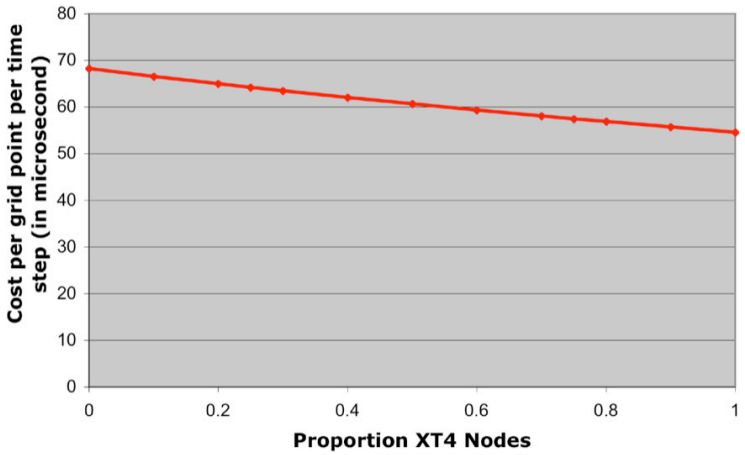


Figure 19: Predicted average cost per grid point when when balancing computational load between Cray XT3 and XT4 nodes on a hybrid system.

mance. Specifically, we can run a smaller problem size on the XT3 nodes than on the XT4 nodes, compensating for the approximately 24% performance degradation caused by the lower memory bandwidth of the XT3 nodes. Overall, we conservatively estimate that a $50 \times 50 \times 40$ grid size on the XT3 nodes will take no longer than the $50 \times 50 \times 50$ problem takes on the XT4 nodes. By only reducing one dimension, we will minimize the coding effort required to support heterogeneous problem sizes per task in S3D. The resulting time per grid point per time step will then depend on the proportion of XT4 nodes to XT3 nodes, as shown in Figure 19. In the current configuration, 46% of the nodes are XT4 nodes, leading to a predicted performance of 61 microseconds per grid point per time step when averaged across all of the processors in the hybrid system.

0.27.2 Improving the node performance of S3D.

Data-intensive codes such as S3D tax the capabilities of microprocessor-based computer systems. The memory subsystem of Jaguar's XT3 compute nodes can deliver less than three bytes per clock cycle from memory to the Opteron processor. For this reason, the speed of computations that do not reuse data in registers or cache is limited by the machine's memory bandwidth; without careful optimization, scientific codes often achieve as little as 5–10% of peak performance on microprocessor-based systems. Therefore, restructuring S3D's data-intensive kernels to reduce their memory bandwidth requirements offers an opportunity for boosting application performance.

Another motivation for restructuring S3D to reduce its memory bandwidth demands is that the XT3 and XT4 nodes in Jaguar differ in the maximum memory bandwidth that they support. Because of their faster memory, the XT4 nodes are 24% faster, as shown in Figure 17. Restructuring S3D to reduce its memory bandwidth needs will reduce the performance disparity between the XT3 and XT4 nodes in the hybrid system.

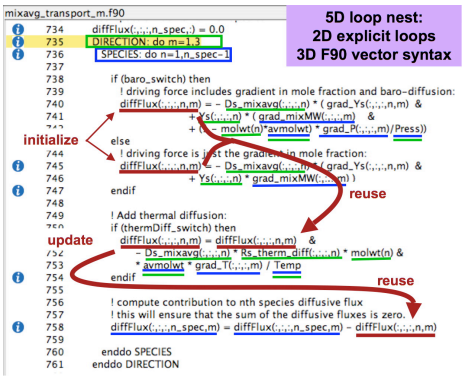


Figure 20: S3D's diffusive flux computation annotated with opportunities for reusing data values.

To investigate the node performance of S3D, we used Rice University's performance tools [246] to study a single-processor execution of a pressure wave test on a 50^3 domain. For the study, we used a single node of a Cray XD1 with a 2.2 GHz Opteron 275 processor and DDR 400 memory, which provides 6.4 GB/s of memory bandwidth (as on Jaguar's XT3 nodes). For the model problem, S3D achieved .305 FLOPs/cycle, which represents 15% of peak.

Using , we pinpointed several data-intensive kernels in S3D that did not fully exploit the memory hierarchy. Figure 20 shows S3D's diffusive flux computation, which was the most costly loop nest in the execution. The loop nest updates a 5-dimensional data structure. Two loops over the direction and the number of species appear explicitly in the source code; other thee-dimensional loops are implicit in the Fortran 90 array operations. For the model problem, this 5D loop nest accounts for 11.3% of S3D's execution time on a Cray XD1 node. By comparing the FLOPs executed by the loop nest and the cycles spent in it, we found that the loop nest achieves only 4% of the theoretical peak performance.

In Figure 20, the code for the diffusive flux computation is overlaid with colored markings that indicate data reuse. Definitions and uses of the `diffFlux` array are underlined in red. The red arrows show how definitions of values are reused by later statements within the DIRECTION and SPECIES loops. The outer loops also offer a myriad of other opportunities for reusing data. References underlined in green will be used by every iteration of the DIRECTION loop since they lack an `m` subscript. References underlined in blue will be used by every iteration of the SPECIES loop since they lack an `n` subscript. If the code is executed as naturally written, almost all of the values will be evicted from cache rather than being reused because each 50^3 slice of the `diffFlux` array almost completely fills the 1MB secondary cache.

To restructure the diffusive flux computation to exploit the potential reuse, we used Rice University's LoopTool utility [247], which supports source-to-source restructuring of loop nests written in Fortran. To apply LoopTool, we outlined the loop nest into a separate file, and marked up the code with directives indicating the transformations

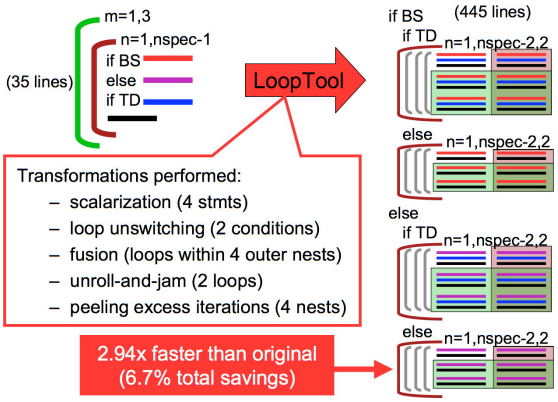


Figure 21: Using LoopTool to optimize S3D's diffusive flux computation on a 50^3 domain.

that it should perform. Figure 21 provides a pictorial rendering of the code restructuring performed with LoopTool. The left side of the figure shows a diagram that represents the structure of the original code. The green and brown arcs represent the DIRECTION and SPECIES loops. The two conditionals are shown explicitly, though the logical predicate variables have been abbreviated. The thick red, purple, blue, and black lines represent the code's Fortran 90 array statements.

The right side of the figure shows a diagram that represents the code after LoopTool applied the transformations listed in the figure. Unswitching the two conditionals out of the original loop nest yields four loop nests, each customized for a particular setting of the switches. The gray arcs in the transformed code represent the loops over the first three dimensions of the `diffFlux` array that arise when LoopTool scalarizes the Fortran 90 array notation. Within each of the four customized loop nests, all of the colored lines representing statement instances have been fused into a single set of triply nested loops. The arc for the green `m` loop is not present in the transformed code. Instead, the code has been unrolled three times; replications of the inner loop resulting from this unrolling are indicated by the green shading. Similarly, the brown `n` loop has been unrolled by two, which causes an orthogonal duplication of each of the three copies of the innermost loop body present after unrolling the `m` loop. This additional duplication is shown by the left-to-right duplication of statements shaded in brown. This figure conveys some of the complexity of the resulting code following LoopTool's transformations. The LoopTool-generated code for this loop nest runs 2.94 times faster than the original. This change alone reduced the entire programs execution time by 6.8% for a 50^3 problem size.

Further analysis of S3D's node performance with , tuning loops with LoopTool, and some manual adjustment of procedure argument passing conventions to avoid copying 4D slices of a 5D array yielded an aggregate improvement of roughly 12.7% for the 50^3 problem size on the Cray XD1.

REPRINT of FY05 S3D INCITE OASCR Joule software assertion

0.28 S3D : A Direct Numerical Simulation of Chemical Combustion

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.

0.28.1 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 [260]. 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.

0.28.2 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 [259, 267, 288, 295]. 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 [285] or unsteady flamelets [287], and conditional moment closure (CMC) [273], those based on the solution of the transport equation for the joint probability density function (PDF) [290], and an approach based on a statistical one-dimensional description of turbulence [272].

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 characterization 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 [267]), 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 [281, 282, 284]. 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.

0.28.3 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.

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 re-ignition 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 chemistry 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 [284], reduced chemistry without heat release and artificial adjustment of rate constants [276] or global one-step chemistry without heat release [292, 279, 280]. 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.

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, [293] 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.

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 [293]. 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 [286, 280]. Multi-dimensional effects can similarly be identified using our parallel surface-based post-processing algorithms [266]. 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 [273], 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 [276]. Differential diffusion represents a challenge for CMC [274, 275], 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 [257, 283] 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 [290]. 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 [279]. 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 [269]. The DNS could assess the importance of the different reignition modes, and provide clues towards empirical models for edge and triple flame propagation.

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 thermo-chemistry, 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.

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.

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 [264]. 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 [270].

0.28.4 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 [266, 259], autoignition [291, 262], to nonpremixed flames [278, 293]. Time advancement is achieved through a six-stage, fourth-order explicit Runge-Kutta (R-K) method [271], spatial differencing is achieved through high-order (eighth-order with tenth-order filters) finite differences on a Cartesian, structured grid [271], and Navier-Stokes Characteristic Boundary Conditions (NSCBC) [289, 294] 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.

0.28.5 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 [277]. 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 [284, 292], 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 [284]. 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 22 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 23 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.

Figure 22: Hydroxyl radical in a turbulent jet flame

Figure 23: Scalar dissipation rate in a turbulent jet flame

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

0.28.6 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 chemistry. 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.

0.28.7 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 0.28.7. 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.

Governing Equations

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

$$\frac{\partial \rho}{\partial t} = -\nabla_\beta \cdot (\rho \mathbf{u}_\beta), \tag{267}$$

$$\frac{\partial (\rho \mathbf{u}_\alpha)}{\partial t} = -\nabla_\beta \cdot (\rho \mathbf{u}_\alpha \mathbf{u}_\beta) + \nabla_\beta \cdot \boldsymbol{\tau}_{\beta\alpha} - \nabla_\alpha p + \rho \sum_{i=1}^{N_s} Y_i \mathbf{f}_{\alpha i}, \tag{268}$$

$$\begin{aligned} \frac{\partial (\rho e_0)}{\partial t} &= -\nabla_\beta \cdot [\mathbf{u}_\beta (\rho e_0 + p)] + \nabla_\beta \cdot (\tau_{\beta\alpha} \cdot \mathbf{u}_\alpha) \\ &\quad -\nabla_\beta \cdot \mathbf{q}_\beta + \rho \sum_{i=1}^{N_s} Y_i \mathbf{f}_{\alpha i} \cdot (\mathbf{V}_{\alpha i} + \mathbf{u}_\alpha), \end{aligned} \tag{269}$$

$$\frac{\partial (\rho Y_i)}{\partial t} = -\nabla_\beta \cdot (\rho Y_i \mathbf{u}_\beta) - \nabla_\beta \cdot (\rho Y_i \mathbf{V}_{\beta i}) + W_i \dot{\omega}_i, \tag{270}$$

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),

$$e_0 = \frac{\mathbf{u}_\alpha \cdot \mathbf{u}_\alpha}{2} - \frac{p}{\rho} + h, \tag{271}$$

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

$$\sum_{i=1}^{N_s} Y_i = 1. \tag{272}$$

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

$$p = \frac{\rho R_u T}{W}, \tag{273}$$

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

$$W = \left(\sum_{i=1}^{N_s} Y_i / W_i \right)^{-1} = \sum_{i=1}^{N_s} X_i W_i. \tag{274}$$

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

$$\frac{Y_i}{X_i} = \frac{W_i}{W}. \tag{275}$$

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.

Constitutive Relationships

The stress tensor, species diffusion velocities, and heat flux vector in equations (268)-(270) are given by [268, 256, 265]

$$\tau_{\beta\alpha} = \tau_{\alpha\beta} = \mu\left[\nabla_\alpha \mathbf{u}_\beta + \nabla_\beta \mathbf{u}_\alpha\right] - \delta_{\alpha\beta}\left(\frac{2}{3}\mu - \kappa\right)\nabla_\gamma \cdot \mathbf{u}_\gamma, \tag{276}$$

$$\mathbf{V}_{\alpha i} = \frac{1}{X_i}\sum_{j=1}^{N_s}\frac{Y_j}{X_j} D_{ij}\mathbf{d}_{\alpha j} - \frac{D_i^T}{\rho Y_i}\nabla_\alpha(\ln T), \tag{277}$$

$$\mathbf{q}_\alpha = -\lambda\nabla_\alpha T + \sum_{i=1}^{N_s} h_i \mathbf{J}_{\alpha i} - \sum_{i=1}^{N_s}\frac{p}{\rho Y_i} D_i^T \mathbf{d}_{\alpha j}. \tag{278}$$

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[268, 256, 265]

$$\mathbf{d}_{\alpha i} = \underbrace{\nabla_\alpha X_i}_{1} + \underbrace{(X_i - Y_i)\nabla_\alpha(\ln p)}_{2} + \underbrace{\frac{\rho Y_i}{p}\left[\mathbf{f}_{\alpha i} - \sum_{j=1}^{N_s} Y_j \mathbf{f}_{\alpha j}\right]}_{3}. \tag{279}$$

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 (279) 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 (276)-(278) in more detail.

Stress Tensor

For monatomic gases, κ is identically zero, and it is often neglected for polyatomic gases as well [268, 256, 265]. It will be neglected in all discussion herein. This simplifies (276) to

$$\tau_{\beta\alpha} = \tau_{\alpha\beta} = \mu \left(\nabla_\alpha \mathbf{u}_\beta + \nabla_\beta \mathbf{u}_\alpha - \frac{2}{3} \delta_{\alpha\beta} \nabla_\gamma \cdot \mathbf{u}_\gamma \right), \tag{280}$$

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

Mass Diffusion Flux

It should be noted that all diagonal components of the multicomponent diffusion coefficient matrix (D_{ii}) are identically zero [268]. Also, the diffusive fluxes and driving forces for all species must sum to zero,

$$\sum_{i=1}^{N_s} \mathbf{J}_{\alpha i} = \sum_{i=1}^{N_s} \rho Y_i \mathbf{V}_{\alpha i} = 0, \qquad \sum_{i=1}^{N_s} \mathbf{d}_{\alpha i} = 0. \tag{281}$$

Equation (277) is often approximated as [268, 256, 296, 265]

$$\mathbf{V}_{\alpha i} = -\frac{D_i^{\text{mix}}}{X_i} \mathbf{d}_{\alpha i} - \frac{D_i^T}{\rho Y_i} \nabla_\alpha \ln T, \tag{282}$$

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

$$D_i^{\text{mix}} = \frac{1 - X_i}{\sum_{j \neq i} X_j / \mathcal{D}_{ij}}, \tag{283}$$

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 (279)) is negligible, then (279) becomes $\mathbf{d}_{\alpha i} = \nabla_\alpha X_i$. If we further neglect the Soret effect, (the second term in (277) and (282)), then (282) reduces to

$$\mathbf{V}_{\alpha i} = -\frac{D_i^{\text{mix}}}{X_i} \nabla_\alpha X_i, \tag{284}$$

which, using (275), can be expressed in terms of mass fractions as

$$\begin{aligned} \mathbf{V}_{\alpha i} &= -\frac{D_i^{\text{mix}}}{Y_i} \left[\nabla_\alpha Y_i + \frac{Y_i}{W} \nabla_\alpha W \right] \\ &= -\frac{D_i^{\text{mix}}}{Y_i} \left[\nabla_\alpha Y_i - Y_i W \sum_{j=1}^{N_s} \frac{\nabla_\alpha Y_j}{W_j} \right]. \end{aligned} \tag{285}$$

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 [263].

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 [268, 261, 296, 263, 265]. In most combustion simulations, the Dufour effect is neglected, and (278) may be written as

$$\mathbf{q}_\alpha = -\lambda \nabla_\alpha T + \sum_{i=1}^{N_s} h_i \mathbf{J}_{\alpha i}. \tag{286}$$

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.

0.28.8 Code Overview

The structure and flow of the code S3D is described in this section and illustrated in Figure 24. 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.

Figure 24: Program flow diagram for S3D

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 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.

0.28.9 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

0.28.10 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.

0.28.11 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 0.28.8.

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. 25. 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.

Figure 25: Performance improvements on Seaborg IBM SP at NERSC

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 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. 25. After these improvements, the code as a whole spends 26% less time on a single processor run. The scalar computing cost was lowered from 1.5x10^{-7}hours/gridpoint/timestep to 1.1x10^{-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.

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. 26. 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 26: 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 $3.55\text{x}10^{-8}$ hours/gridpt/timestep. As seen in Fig. 26 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.

0.28.12 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.

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 27 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 27: Code scaling on IBM SP

Scaling on Cray X1

Preliminary scaling tests on 1,8,64, and 256 processors have been performed on the Cray X1. Figure 28 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 28: Code scaling on Cray X1

0.28.13 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.

0.28.14 Q3 Progress and Dynamics

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%.

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.

0.28.15 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 0.28.5. 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.

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 [298]. A key element in the PDF approach is a mixing time scale. Pope [299] 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.

Mathematical definition

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

$$\tau_\phi = \frac{\overline{\phi'^2}}{\overline{\chi_\phi}}, \tag{287}$$

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

$$\chi_\phi = 2D_\phi \nabla\phi \cdot \nabla\phi, \tag{288}$$

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:

$$\tau_u = \frac{\overline{k}}{\overline{\epsilon}}, \tag{289}$$

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:

$$r_\phi = \frac{\int_0^{\delta_Z} \tau_u dy}{\int_0^{\delta_Z} \tau_\phi dy}, \tag{290}$$

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.

Effect of Schmidt Number

Figure 29 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 to values reported by experiments,

Figure 29: Mechanical to scalar timescale ratio for mixture fraction versus time.

simple chemistry DNS and used successfully in modelling [301]. 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 30 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 30: Mechanical to scalar timescale ratio for H, H_2 and CO_2 mass fractions and mixture fraction versus time.

Effect of Reaction-Diffusion coupling

The strong interplay between reaction and diffusion in nonpremixed flames can also affect mixing timescales. Figure 31 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 32a) shows the HO_2 mass fraction, and Figure 32b) 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 33 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 34 obtained from the calculations performed on the Cray X1 and X1E show similar trends as reported above for the PNNL data.

Figure 31: Mechanical to scalar timescale ratio for HO_2, H_2O_2, O and OH mass fractions and mixture fraction versus time.

Figure 32: a) HO2 mass fraction on a color scale b) OH mass fraction on a color scale. White contours of χ.

Figure 33: Iso-surfaces of dissipation rates. Blue: χ. Red: χ_{HO2}. Green: χ_{OH}.

Figure 34: (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)

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 [300]. 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.

0.28.16 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. 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.

0.28.17 Prospects for future combustion simulations on petascale platforms

The computational cost of a DNS is largely determined by the spatial and temporal resolution requirements. The resolution requirements in a reactive 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 0.28.7. 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.

0.28.18 Code Improvements made in FY05-Q4

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%.

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[35].

Figure 35: Code scaling on various platforms as of Q4-2005.

Figure 36: Execution time versus number of processors on various platforms as of Q4-2005.

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[36].

Bibliography

[1] Akiyama, S., Wheeler, J. C., Meier, D. L., & Lichtenstadt, I. 2003, ApJ, 584, 954

[2] Angulo, C., Arnould, M., Rayet, M., Descouvemont, P., Baye, D., Leclercq-Willain, C., Coc, A., Barhoumi, S., Aguer, P., Rolfs, C., Kunz, R., Hammer, J. W., Mayer, A., Paradellis, T., Kossionides, S., Chronidou, C., Spyrou, K., degl'Innocenti, S., Fiorentini, G., Ricci, B., Zavatarelli, S., Providencia, C., Wolters, H., Soares, J., Grama, C., Rahighi, J., Shotter, A., & Lamehi Rachti, M. 1999, Nucl. Phys. A, 656, 3

[3] Argast, D., Samland, M., Thielemann, F.-K., & Gerhard, O. E. 2002, A&A, 388, 842

[4] Argast, D., Samland, M., Thielemann, F.-K., & Qian, Y.-Z. 2004, A&A, 416, 997

[5] Arnett, W. & Truran, J. 1969, ApJ, 157, 339

[6] Arnett, W. D. 1996, Supernovae and nucleosynthesis : an investigation of the history of matter, from the big bang to the present (Princeton: Princeton University Press)

[7] Audi, G. & Wapstra, A. H. 1995, Nucl. Phys. A, 595, 409

[8] Baker, J. G., Centrella, J., Choi, D.-I., Koppitz, M., & van Meter, J. 2006, Phys. Rev. D, 73, 104002

[9] Bao, Z. Y., Beer, H., Käppeler, F., Voss, F., Wisshak, K., & Rauscher, T. 2000, At. Data Nuc. Data Tab., 76, 70

[10] Baron, E., Myra, E. S., Cooperstein, J., & van den Horn, L. J. 1989, ApJ, 339, 978

[11] Baumgarte, T. W. & Shapiro, S. L. 1998, Phys. Rev. D, 59, 024007

[12] Bazan, G. & Arnett, W. D. 1998, ApJ, 496, 316

[13] Benz, W., Thielemann, F.-K., & Hills, J. G. 1989, ApJ, 342, 986

[14] Bethe, H. A. & Wilson, J. R. 1985, ApJ, 295, 14

[15] Blondin, J., Mezzacappa, A., & DeMarino, C. 2003, ApJ, 584, 971

[16] Blondin, J. M. 2005a, in Proceedings from the Institute for Nuclear Theory, Vol. 14, Open Issues in Core Collapse Supernova Theory, ed. A. Mezzacappa & G. M. Fuller (Singapore: World Scientific), 123

[17] Blondin, J. M. 2005b, Journal of Physics Conference Series, 16, 370

[18] Blondin, J. M. & Lundqvist, P. 1993, ApJ, 405, 337

[19] Blondin, J. M. & Mezzacappa, A. 2007, Nature, 445, 58

[20] Blondin, J. M. & Shaw, S. 2007, ApJ, 656, 366

[21] Bodansky, D., Clayton, D., & Fowler, W. 1968, ApJS, 16, 299

[22] Borzov, I. N. 2006, Nucl. Phys. A, 777, 645

[23] Brown, L. S. & Sawyer, R. F. 1997, ApJ, 489, 968

[24] Bruenn, S., Raley, E., & Mezzacappa, A. 2004, ApJ, in press

[25] Bruenn, S. W. 1985, ApJS, 58, 771

[26] Bruenn, S. W., De Nisco, K. R., & Mezzacappa, A. 2001, ApJ, 560, 326

[27] Bruenn, S. W., Dirk, C. J., Mezzacappa, A., Hayes, J. C., Blondin, J. M., Hix, W. R., & Messer, O. E. B. 2006, Journal of Physics: Conference Series, 46, 393

[28] Buchmann, L. R. & Barnes, C. A. 2006, Nucl. Phys. A, 777, 254

[29] Buras, R., Janka, H.-T., Rampp, M., & Kifonidis, K. 2006a, A&A, 457, 281

[30] Buras, R., Rampp, M., Janka, H.-T., & Kifonidis, K. 2006b, A&A, 447, 1049

[31] Burrows, A., Dessart, L., Livne, E., Ott, C. D., & Murphy, J. 2007, ArXiv Astrophysics e-prints

[32] Burrows, A. & Goshy, J. 1993, ApJ, 416, L75

[33] Burrows, A., Hayes, J., & Fryxell, B. A. 1995, ApJ, 450, 830

[34] Burrows, A. & Lattimer, J. M. 1985, ApJ, 299, L19

[35] Burrows, A., Livne, E., Dessart, L., Ott, C. D., & Murphy, J. 2006, ApJ, 640, 878

[36] Campanelli, M., Lousto, C. O., Marronetti, P., & Zlochower, Y. 2006, Phys. Rev. Lett., 96, 111101

[37] Castor, J. I. 1972, ApJ, 178, 779

[38] Cayrel, R., Depagne, E., Spite, M., Hill, V., Spite, F., François, P., Plez, B., Beers, T., Primas, F., Andersen, J., Barbuy, B., Bonifacio, P., Molaro, P., & Nordström, B. 2004, A&A, 416, 1117

[39] Cerdá-Durán, P., Faye, G., Dimmelmeier, H., Font, J. A., Ibáñez, J. M., Müller, E., & Schäfer, G. 2005, A&A, 439, 1033

[40] Chieffi, A. & Limongi, M. 2002, ApJ, 577, 281

[41] —. 2004, ApJ, 608, 405

[42] Christy, R. F. 1964, Rev. Mod. Phys., 36, 555

[43] Clayton, D. D. 1983, Principles of Stellar Evolution and Nucleosynthesis (Chicago: University of Chicago Press)

[44] Colella, P. & Woodward, P. 1984, J. Comput. Phys., 54, 174

[45] Cooperstein, J. 1985, Nucl. Phys. A, 438, 722

[46] D'Azevedo, E. F., Messer, B., Mezzacappa, A., & Liebendörfer, M. 2005, SIAM J. Sci. Comp., 26, 810

[47] Descouvemont, P. & Rauscher, T. 2006, Nucl. Phys. A, 777, 137

[48] Dewitt, H. E., Graboske, H. C., & Cooper, M. S. 1973, ApJ, 181, 439

[49] Dimmelmeier, H., Font, J. A., & Müller, E. 2002a, A&A, 388, 917

[50] —. 2002b, A&A, 393, 523

[51] Falk, S. W. & Arnett, W. D. 1977, ApJS, 33, 515

[52] Fowler, W., Caughlan, G., & Zimmermann, B. 1967, ARA&A, 5, 525

[53] Frebel, A., Aoki, W., Christlieb, N., Ando, H., Asplund, M., Barklem, P. S., Beers, T. C., Eriksson, K., Fechner, C., Fujimoto, M. Y., Honda, S., Kajino, T., Minezaki, T., Nomoto, K., Norris, J. E., Ryan, S. G., Takada-Hidai, M., Tsangarides, S., & Yoshii, Y. 2005, Nature, 434, 871

[54] Freiburghaus, C., Hix, W. R., & Thielemann, F.-K. 1999, in Proceeding of Nuclei in the Cosmos V, ed. N. Prantzos (Paris:Editions Frontiere), 140

[55] Fröhlich, C., Hauser, P., Liebendörfer, M., Martínez-Pinedo, G., Thielemann, F.-K., Bravo, E., Zinner, N. T., Hix, W. R., Langanke, K., Mezzacappa, A., & Nomoto, K. 2006a, ApJ, 637, 415

[56] Fröhlich, C., Martínez-Pinedo, G., Liebendörfer, M., Thielemann, F. K., Bravo, E., Hix, W. R., Langanke, K., & Zinner, N. T. 2006b, Phys. Rev. Lett., 96, 142502

[57] Fryer, C. L. 1999, ApJ, 522, 413

[58] Fryer, C. L. & Heger, A. 2000, ApJ, 541, 1033

[59] Fryer, C. L. & Warren, M. S. 2002, ApJ, 574, L65

[60] —. 2004, ApJ, 601, 391

[61] Fryxell, B., Müller, E., & Arnett, W. 1989, Hydrodynamics and Nuclear Burning, Tech. rep., Max-Planck-Institut für Astophysik, Garching, Preprint 449

[62] Fuller, G. M. & Meyer, B. S. 1995, ApJ, 453, 792

[63] Galama, T. J., Vreeswijk, P. M., van Paradijs, J., Kouveliotou, C., Augusteijn, T., Bohnhardt, H., Brewer, J. P., Doublier, V., Gonzalez, J.-F., Leibundgut, B., Lidman, C., Hainaut, O. R., Patat, F., Heise, J., in 't Zand, J., Hurley, K., Groot, P. J., Strom, R. G., Mazzali, P. A., Iwamoto, K., Nomoto, K., Umeda, H., Nakamura, T., Young, T. R., Suzuki, T., Shigeyama, T., Koshut, T., Kippen, M., Robinson, C., de Wildt, P., Wijers, R. A. M. J., Tanvir, N., Greiner, J., Pian, E., Palazzi, E., Frontera, F., Masetti, N., Nicastro, L., Feroci, M., Costa, E., Piro, L., Peterson, B. A., Tinney, C., Boyle, B., Cannon, R., Stathakis, R., Sadler, E., Begam, M. C., & Ianna, P. 1998, Nature, 395, 670

[64] Gardiner, T. A. & Stone, J. M. 2005, J. Comp. Phys., 205, 509

[65] Gear, C. 1971, Numerical Initial Value Problems in Ordinary Differential Equations (Englewood Cliffs, NJ: Prentice-Hall)

[66] Hanhart, C., Phillips, D. R., & Reddy, S. 2001, Phys. Lett. B, 499, 9

[67] Hannestad, S. & Raffelt, G. 1998, ApJ, 507, 339

[68] Haxton, W. C., Parker, P. D., & Rolfs, C. E. 2006, Nucl. Phys. A, 777, 226

[69] Herant, M., Benz, W., Hix, W. R., Fryer, C. L., & Colgate, S. A. 1994, ApJ, 435, 339

[70] Hix, W. R., Khokhlov, A. M., Wheeler, J. C., & Thielemann, F.-K. 1998, ApJ, 503, 332

[71] Hix, W. R., Parete-Koon, S., Freiburghaus, C., & Thielemann, F.-K. 2007, ApJ, submitted

[72] Hix, W. R. & Thielemann, F.-K. 1996, ApJ, 460, 869

[73] Hix, W. R. & Thielemann, F.-K. 1999, ApJ, 511, 862

[74] Hix, W. R. & Thielemann, F.-K. 1999, ApJ, 511, 862

[75] Honda, S., Aoki, W., Ando, H., Izumiura, H., Kajino, T., Kambe, E., Kawanomoto, S., Noguchi, K., Okita, K., Sadakane, K., Sato, B., Takada-Hidai, M., Takeda, Y., Watanabe, E., Beers, T. C., Norris, J. E., & Ryan, S. G. 2004, ApJS, 152, 113

[76] Ichimaru, S. 1993, Reviews of Modern Physics, 65, 255

[77] Janka, H.-T. 2001, A&A, 368, 527

[78] Janka, H.-T., Buras, R., Kitaura Joyanes, F. S., Marek, A., Rampp, M., & Scheck, L. 2005, Nucl. Phys. A, 758, 19

[79] Janka, H.-T. & Müller, E. 1996, A&A, 306, 167

[80] Juodoagalvis, A., Langanke, K., Martínez-Pinedo, G., Hix, W. R., Dean, D. J., & Sampaio, J. M. 2005, Nucl. Phys. A, 747, 87

[81] Kageyama, A. & Sato, T. 2004, Geochemistry, Geophysics, Geosystems, 5, 9

[82] Käppeler, F. & Mengoni, A. 2006, Nucl. Phys. A, 777, 291

[83] Käppeler, F., Thielemann, F.-K., & Wiescher, M. 1998, Annu. Rev. Nucl. Part. Sci., 48, 175

[84] Khokhlov, A. 1996, YASS - Yet another stiff solver that works., Tech. rep., NRL Technical Memorandum Report

[85] Khokhlov, A. M. 1995, ApJ, 449, 695

[86] Kifonidis, K., Plewa, T., Janka, H.-T., & Müller, E. 2003, A&A, 408, 621

[87] Kutter, G. S. & Sparks, W. M. 1972, ApJ, 175, 407

[88] Lamb, D. Q., Lattimer, J. M., Pethick, C. J., & Ravenhall, D. G. 1978, Phys. Rev. Lett., 41, 1623

[89] Lambert, J. 1980, in Computational Techniques for Ordinary Differential Equations, ed. I. Gladwell & D. Sayars (New York: Academic Press), 19–46

[90] LARROUTUROU, B. 1991, J. Comp. Phys., 95, 59

[91] Lattimer, J. & Swesty, F. D. 1991, Nucl. Phys. A, 535, 331

[92] Lattimer, J. M., Pethick, C. J., Ravenhall, D. G., & Lamb, D. Q. 1985, Nucl. Phys. A, 432, 646

[93] Ledoux, P. & Walraven, T. 1958, Handbuch der Physik, 51, 353

[94] Liebendörfer, M., Messer, O. E. B., Mezzacappa, A., Bruenn, S. W., Cardall, C. Y., & Thielemann, F.-K. 2004, ApJS, 150, 263

[95] Liebendörfer, M., Rampp, M., Janka, H.-T., & Mezzacappa, A. 2005, ApJ, 620, 840

[96] Lindquist, R. W. 1966, Annals of Physics, 37, 487

[97] Marcucci, L. E., Nollett, K. M., Schiavilla, R., & Wiringa, R. B. 2006, Nucl. Phys. A, 777, 111

[98] Marek, A., Dimmelmeier, H., Janka, H.-T., Müller, E., & Buras, R. 2006, A&A, 445, 273

[99] Martínez-Pinedo, G., Liebendörfer, M., & Frekers, D. 2006, Nucl. Phys. A, 777, 395

[100] Matheson, T., Garnavich, P. M., Stanek, K. Z., Bersier, D., Holland, S. T., Krisciunas, K., Caldwell, N., Berlind, P., Bloom, J. S., Bolte, M., Bonanos, A. Z., Brown, M. J. I., Brown, W. R., Calkins, M. L., Challis, P., Chornock, R., Echevarria, L., Eisenstein, D. J., Everett, M. E., Filippenko, A. V., Flint, K., Foley, R. J., Freedman, D. L., Hamuy, M., Harding, P., Hathi, N. P., Hicken, M., Hoopes, C., Impey, C., Jannuzi, B. T., Jansen, R. A., Jha, S., Kaluzny, J., Kannappan, S., Kirshner, R. P., Latham, D. W., Lee, J. C., Leonard, D. C., Li, W., Luhman, K. L., Martini, P., Mathis, H., Maza, J., Megeath, S. T., Miller, L. R., Minniti, D., Olszewski, E. W., Papenkova, M., Phillips, M. M., Pindor, B., Sasselov, D. D., Schild, R., Schweiker, H., Spahr, T., Thomas-Osip, J., Thompson, I., Weisz, D., Windhorst, R., & Zaritsky, D. 2003, ApJ, 599, 394

[101] Meakin, C. A. & Arnett, D. 2006, ApJ, 637, L53

[102] Messer, O. E. B., Mezzacappa, A., Bruenn, S. W., & Guidry, M. W. 1998, ApJ, 507, 353

[103] Meyer, B. S., Krishnan, T. D., & Clayton, D. D. 1998, ApJ, 498, 808

[104] Mezzacappa, A. 2005, Annu. Rev. Nucl. Part. Sci., 55, 467

[105] Mezzacappa, A., Bruenn, S. W., Blondin, J. M., Hix, W. R., & Bronson Messer, O. E. 2007, in American Institute of Physics Conference Series, Vol. 924, THE MULTICOLORED LANDSCAPE OF COMPACT OBJECTS AND THEIR EXPLOSIVE ORIGINS, 234–242

[106] Mezzacappa, A., Calder, A. C., Bruenn, S. W., Blondin, J. M., Guidry, M. W., Strayer, M. R., & Umar, A. S. 1998, ApJ, 495, 911

[107] Mezzacappa, A., Liebendörfer, M., Messer, O., Hix, W., Thielemann, F.-K., & Bruenn, S. 2001, Phys. Rev. Lett., 86, 1935

[108] Mezzacappa, A. & Matzner, R. A. 1989, ApJ, 343, 853

[109] Mezzacappa, A. & Messer, O. E. B. 1999, J. Comp. Appl. Math, 109, 281

[110] Mihalas, D. & Mihalas, B. 1984, Foundations of Radiation Hydrodynamics (New York: Oxford University Press)

[111] Miralles, J. A., Pons, J. A., & Urpin, V. A. 2002, ApJ, 574, 356

[112] Mohr, P., Zilges, A., & Utsonomi. 2004, Nucl. Phys. A, this volume

[113] Möller, P., Nix, J. R., & Kratz, K.-L. 1997, At. Data Nuc. Data Tab., 66, 131

[114] Mott, D., Oran, E., & van Leer B. 2000, J. Comp. Phys., 164, 407

[115] Müller, E. 1986, A&A, 162, 103

[116] Müller, E. 1998, in Saas-Fee Advanced Course 27: Computational Methods for Astrophysical Fluid Flow., 343

[117] Müller, E. & Steinmetz, M. 1995, Computer Physics Communications, 89, 45

[118] Nagataki, S., Shimizu, T. M., & Sato, K. 1998, ApJ, 495, 413

[119] Nakamura, T., Umeda, H., Iwamoto, K., Nomoto, K., Hashimoto, M.-a., Hix, W. R., & Thielemann, F.-K. 2001, ApJ, 555, 880

[120] Nakamura, T., Umeda, H., Nomoto, K., Thielemann, F.-K., & Burrows, A. 1999, ApJ, 517, 193

[121] Niemeyer, J. C. & Hillebrandt, W. 1995, ApJ, 452, 769

[122] Nomoto, K., Hashimoto, M., Tsujimoto, T., Thielemann, F.-K., Kishimoto, N., Kubo, Y., & Nakasato, N. 1997, Nucl. Phys. A, 616, 79

[123] Ohnishi, N., Kotake, K., & Yamada, S. 2006, ApJ, 641, 1018

[124] Oran, E. & Boris, J. 2001, Numerical Simulation of Reactive Flow, 2nd Edition (Cambridge: Cambridge University Press)

[125] Plewa, T. & Müller, E. 1999, A&A, 342, 179

[126] Prantzos, N., Arnould, M., & Arcoragi, J.-P. 1987, ApJ, 315, 209

[127] Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. 1992, Numerical Recipes ((Cambridge:Cambridge Univ.)), second edition

[128] Pruet, J., Hoffman, R. D., Woosley, S. E., Janka, H.-T., & Buras, R. 2006, ApJ, 644, 1028

[129] Pruet, J., Woosley, S. E., Buras, R., Janka, H.-T., & Hoffman, R. D. 2005, ApJ, 623, 325

[130] Rampp, M. & Janka, H.-T. 2002, A&A, 396, 361

[131] Rauscher, T. 2003, ApJS, 147, 403

[132] Rauscher, T., Heger, A., Hoffman, R. D., & Woosley, S. E. 2002, ApJ, 576, 323

[133] Rauscher, T. & Thielemann, F. 2000, At. Data Nuc. Data Tab., 75, 1

[134] Rauscher, T., Thielemann, F.-K., & Kratz, K.-L. 1996, MmRAS, 67, 851

[135] Ravenhall, D. G., Pethick, C. J., & Wilson, J. R. 1983, Phys. Rev. Lett., 50, 2066

[136] Reddy, S., Prakash, M., & Lattimer, J. M. 1998, Phys. Rev. D, 58, 013009

[137] Rolfs, C. & Rodney, W. 1988, Cauldrons in the Cosmos (Chicago:Univ. of Chicago)

[138] Salpeter, E. E. 1954, Australian Journal of Physics, 7, 373

[139] Scheck, L., Kifonidis, K., Janka, H.-T., & Müller, E. 2006, A&A, 457, 963

[140] Shen, H., Toki, H., Oyamatsu, K., & Sumiyoshi, K. 1998, Prog. Theor. Phys., 100, 1013

[141] Shibata, M. & Nakamura, T. 1995, Phys. Rev. D, 52, 5428

[142] Smarr, L., Wilson, J. R., Barton, R. T., & Bowers, R. L. 1981, ApJ, 246, 515

[143] Sneden, C., Cowan, J. J., Lawler, J. E., Ivans, I. I., Burles, S., Beers, T. C., Primas, F., Hill, V., Truran, J. W., Fuller, G. M., Pfeiffer, B., & Kratz, K. 2003, ApJ, 591, 936

[144] Symbalisty, E. M. D. 1984, ApJ, 285, 729

[145] Thielemann, F.-K., Nomoto, K., & Hashimoto, M. 1996, ApJ, 460, 408

[146] Timmes, F. X. 1999, ApJS, 124, 241

[147] Timmes, F. X., Hoffman, R. D., & Woosley, S. E. 2000, ApJS, 129, 377

[148] Umeda, H. & Nomoto, K. 2002, ApJ, 565, 385

[149] —. 2005, ApJ, 619, 427

[150] Weaver, T. A., Zimmerman, G. B., & Woosley, S. E. 1978, ApJ, 225, 1021

[151] Wilson, J. R. 1985, in Numerical Astrophysics, ed. J. M. Centrella, J. M. LeBlanc, & R. L. Bowers (Boston: Jones and Bartlett), 422–434

[152] Wilson, J. R. & Mathews, G. J. 1995, Phys. Rev. Lett., 75, 4161

[153] Wilson, J. R., Mathews, G. J., & Marronetti, P. 1996, Phys. Rev. D, 54, 1317

[154] Wilson, J. R. & Mayle, R. W. 1993, Phys. Rep., 227, 97

[155] Woosley, S. & Janka, H.-T. 2005, Nature Phys., 1, 147

[156] Woosley, S. E., Arnett, W. D., & Clayton, D. D. 1973, ApJS, 26, 231

[157] Woosley, S. E. & Weaver, T. A. 1995, ApJS, 101, 181

[158] W. M. Tang, Nucl. Fusion **18**, 1089 (1978).

[159] J. W. Connor and H. R. Wilson, Plasma Phys. Control. Fusion **36**, 719 (1994).

[160] W. Horton, Rev. Mod. Phys. **71**, 735 (1999).

[161] S. Parker, W. Lee, and R. Santoro, Phys. Rev. Lett. **71**, 2042 (1993).

[162] M. Kotschenreuther, G. Rewoldt, and W. M. Tang, Comput. Phys. Commun. **88**, 128 (1995).

[163] R. D. Sydora, V. K. Decyk, and J. M. Dawson, Plasma Phys. Control. Fusion **38**, A281 (1996).

[164] M. A. Beer and G. W. Hammett, Phys. Plasmas **3**, 4046 (1996).

[165] Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang and R. White, Science **281**, 1835 (1998).

[166] W. Dorland, *Proc. 18th Intnl. Conf. on Fusion Energy, Sorrento, Italy, 2000*, (International Atomic Energy Agency).

[167] A. M. Dimits, G. Bateman, M. A. Beer et al., Phys. Plasmas **7**, 969 (2000).

[168] F. Jenko, Comput. Phys. Commun. **125**, 196 (2000).

[169] J. Candy and R. Waltz, J. Comput. Phys. **186**, 545 (2003).

[170] Y. Chen and S. Parker, J. Comput. Phys. **189**, 463 (2003).

[171] Y. Idomura, S. Tokuda and Y. Kishimoto, Nucl. Fusion **43**, 243 (2003).

[172] L. Villard et al., Nucl. Fusion **44**, 172 (2004).

[173] W. W. Lee, Phys. Fluids **26**, 556 (1983).

[174] Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and P. H. Diamond, Phys. Rev. Lett. **83**, 3645 (1999).

[175] Z. Lin, S. Ethier, T. S. Hahm, and W. M. Tang, Phys. Rev. Lett. **88**, 1950004 (2002).

[176] Z. Lin and T. S. Hahm, Phys. Plasmas **11**, 1099 (2004).

[177] T. S. Hahm, P. H. Diamond, Z. Lin, K. Itoh, and S.-I. Itoh, Plasma Phys. Control. Fusion **46**, 323 (2004).

[178] T. S. Hahm, P. H. Diamond, Z. Lin, G. Rewoldt, O. Gurcan, and S. Ethier, Phys. Plasmas **12**, 090903 (2005).

[179] R. J. Goldston, *Basic Physical Processes of Toroidal Fusion Plasmas*, Proceedings of Course and Workshop, Varenna, 1985, edited by G. P. Lampis, M. Lontano, G. G. Leotta, A. Malein, and E. Sindoni (Monotypia Franchi, Citta di Castello, 1985), Vol. 1, p. 165.

[180] J. DeLucia, S. C. Jardin, and A. M. M. Todd, J. Comput. Phys. **37**, 183 (1980).

[181] L. E. Zakharov and A. Pletzer, Phys. Plasmas **6**, 4693 (1999).

[182] J. Luxon, Nucl. Fusion **42**, 614 (2002).

[183] M. Ono, S. M. Kaye, Y.-K. M. Peng *et al.*,Nucl. Fusion **40**, 557 (2000).

[184] B. Scott, Phys. Plasmas **6**, 2334 (1998).

[185] R. White and L. E. Zakharov, Phys. Plasmas **10**, 573 (2003).

[186] R. J. Littlejohn, Phys. Fluids **24**, 1730 (1981).

[187] E. A. Frieman and L. Chen, Phys. Fluids **25**, 502 (1982).

[188] D. H. E. Dubin, J. A. Krommes, C. Oberman, and W. W. Lee, Phys. Fluids **26**, 3524 (1983).

[189] T. S. Hahm, Phys. Fluids **31**, 2670 (1988).

[190] A. J. Brizard, J. Plasma Phys. **41**, 541 (1989).

[191] T. S. Hahm, Phys. Plasmas **3**, 4658 (1996).

[192] W. W. Lee, J. Comput. Phys. **72**, 243 (1987).

[193] Z. Lin and W. W. Lee, Phys. Rev. E **52**, 5646 (1995).

[194] F. L. Hinton and S. K. Wong, Phys. Fluids **28**, 3028 (1985).

[195] W. X. Wang, F. L. Hinton, and K. Wang, Phys. Rev. Lett. **87**, 055002 (2001).

[196] W. X. Wang, W. M. Tang, F. L. Hinton, L. E. Zakharov, R. B. White, and J. Manickam, Comput. Phys. Commun. **164**, 178 (2004).

[197] Z. Lin and L. Chen, Phys. Plasmas **8**, 1447 (2001).

[198] W. X. Wang, L. Chen, and Z. Lin, Bull. Am. Phys. Soc. **46**(7), 114 (2001).

[199] I. Manuilskiy and W. W. Lee, Phys. Plasmas **7** 1381 (2000).

[200] J. L. V. Lewandowski, Phys. Plasmas **10**, 3204 (2003).

[201] Y. Nishimura, Z. Lin, J. L. V. Lewandowski, and S. Ethier, J. Comput. Phys. (to be published).

[202] J. V. W. Reynders, Ph.D. Thesis, Princeton University (1992).

[203] Z. Lin, L. Chen, Y. Nishimura, H. Qu, T.S. Hahm, J. Lewandowski, G. Rewoldt, W.X. Wang, P.H. Diamond, C. Holland, F. Zonca, Y. Li, Paper TH/8-4, Proceedings of the 20th IAEA Fusion Energy Conference, Villamoura (2004). [http://www-naweb.iaea.org/napc/physics/fec/fec2004/papers/th_8-4.pdf]

[204] S. Hodson, Using Lustre Effectively" NCCS Scaling Workshop, August 1, 2007, Oak Ridge, TN.

[205] Abbasi, H., Wolf, M., and Schwan, K. "LIVE: a light-weight data workspace for computational science." In Proceedings of the 16th international Symposium on High Performance Distributed Computing.

[206] Docan, C., Parashar, M., Klasky, S., "High Speed Asynchronous Data Transfer on the Cray XT3", Cray User's Group, May 7-10, 2007, Seattle, WA.

[207] P. H. Diamond, K. Itoh, S.-I. Itoh, and T. S. Hahm, Plasma Phys. Control. Fusion **47**, R35 (2005).

[208] F. L. Hinton and M. N. Rosenbluth, Plasma Phys. Control. Fusion **41**, A653 (1999).

[209] M. A. Malkov, P. H. Diamond, and M. N. Rosenbluth, Phys. Plasmas **8**, 5073 (2001).

[210] R. White, L. Chen, and F. Zonca, Phys. Plasmas **12**, 057304 (2005).

[211] B. N. Rogers, W. Dorland, and M. Kotschenreuther, Phys. Rev. Lett. **85**, 5336 (2000).

[212] E. J. Kim and P. H. Diamond, Phys. Plasmas **9**, 4530 (2002).

[213] N. Miyato, Y. Kishimoto, and J. Li, Phys. Plasmas **12**, 5557 (2004).

[214] R. Nazikian, K. Shinohara, G. J. Kramer et al., Phys. Rev. Lett. **94**, 135002 (2005).

[215] N. Peters. *Turbulent Combustion*. Cambridge university press, New York, 2000.

[216] S. A. Filatyev, J. F. Driscoll, C. D. Carter, and J. M. Donbar. Measured properties of turbulent premixed flames for model assessment, including burning velocities, stretch rates, and surface densities. *Combustion and Flame*, 141:1–21, 2005.

[217] R. J. Kee, J. F. Grcar, M. D. Smooke, and J. A. Miller. PREMIX: A fortran program for modeling steady laminar one-dimensional premixed flames. Technical Report SAND85-8240, Sandia National Laboratories, 1985.

[218] T. Plessing, N. Peters, and J. G. Wunning. Laser optical investigation of highly preheated combustion with strong exhaust gas recirculation. *Proceedings of the Combustion Institute*, 27:3197–3204, 1998.

[219] R. Sankaran, E. R. Hawkes, J. H. Chen, T. Lu, and C. K. Law. Study of turbulent premixed flame thickness using direct numerical simulations in a slot burner configuration. In *Proceedings of the 44nd Aerospace Sciences Meeting and Exhibit, Reno, USA*, 2006. AIAA Paper 2006-0165.

[220] R. Sankaran, E. R. Hawkes, J. H. Chen, T. Lu, and C. K. Law. Structure of a spatially-developing turbulent lean methane-air bunsen flame. *Proceedings of the Combustion Institute*, 31:1291–1298, 2007.

[221] M. Frenklach, H. Wang, M. Goldenberg, G. P. Smith, D. M. Golden, C. T. Bowman, R. K. Hanson, W. C. Gardiner, and V. Lissianski. GRI-Mech: An optimized detailed chemical reaction mechanism for methane combustion. Technical Report GRI-95/0058, Gas Research Institute, 1995.

[222] T. Lu and C. K. Law. A directed relation graph method for mechanism reduction. *Proceedings of the Combustion Institute*, 30:1333–1341, 2005.

[223] T. Lu, Y. Ju, and C. K. Law. Complex CSP for chemistry reduction and analysis. *Combustion and Flame*, 126:1445–1455, 2001.

[224] C. A. Kennedy and M. H. Carpenter. Several new numerical methods for compressible shear-layer simulations. *Applied Numerical Mathematics*, 14(4):367–458, 1994.

[225] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis. Low-storage explicit runge-kutta schemes for the compressible navier-stokes equations. *Applied Numerical Mathematics*, 35(3):177–264, 2000.

[226] R. J. Kee, F. M. Rupley, and J. A. Miller. Chemkin-II: A fortran chemical kinetics package for the analysis of gas-phase chemical kinetics. Technical Report SAND89-8009B, Sandia National Laboratories, 1991.

[227] T. J. Poinsot and S. K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics*, 101:104–129, 1992.

[228] J. C. Sutherland and C. A. Kennedy. Improved boundary conditions for viscous, reacting, compressible flows. *Journal of Computational Physics*, 191:502–524, 2003.

[229] C. S. Yoo, Y. Wang, A. Trouve, and H. G. Im. Characteristic boundary conditions for direct simulations of turbulent counterflow flames. *Combustion Theory and Modelling*, 9:617–646, 2005.

[230] C. S. Yoo and H. G. Im. Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects. *Combustion Theory and Modelling*, 11(2):259–286, 2007.

[231] N. Peters. The turbulent burning velocity for large-scale and small-scale turbulence. *Journal of Fluid Mechanics*, 384:107–132, 1999.

[232] M. S. Mansour, N. Peters, and Y. C. Chen. Investigation of scalar mixing in the thin reaction zones regime using a simultaneous CH-LIF/rayleigh laser technique. *Proceedings of the Combustion Institute*, 27:767–773, 1998.

[233] Y. C. Chen and M. S. Mansour. Investigation of flame broadening in turbulent premixed flames in the thin reaction zones regime. *Proceedings of the Combustion Institute*, 27:811–818, 1998.

[234] F. Dinkelacker, A. Soika, D. Most, D. Hofmann, A. Leipertz, W. Polifke, and K. Dobbeling. Structure of locally quenched highly turbulent lean premixed flames. *Proceedings of the Combustion Institute*, 27:857–865, 1998.

[235] A. Soika, F. Dinkelacker, and A. Leipertz. Measurement of the resolved flame structure of turbulent premixed flames with constant reynolds number and varied stoichiometry. *Proceedings of the Combustion Institute*, 27:785–792, 1998.

[236] F. Dinkelacker. Experimental validation of flame regimes for highly turbulent premixed flames. In *Proceedings of the European Combustion Meeting*, 2003. Paper no. 158.

[237] E. R. Hawkes and J. H. Chen. Direct numerical simulation of premixed flames in the thin reaction zones regime. In *Proceedings of the 2003 Winter Meeting of the Western States Section of the Combustion Institute*, 2003. Paper no. 03F-68.

[238] D. Thevenin. Three-dimensional direct simulations and structure of expanding turbulent methane flames. *Proceedings of the Combustion Institute*, 30:629–637, 2005.

[239] R. Sankaran, E. R. Hawkes, C. S. Yoo, J. H. Chen, T. Lu, and C. K. Law. Direct numerical simulation of stationary lean premixed methane-air flames under intense turbulence. In *Proceedings of the Fifth US Combustion Meeting*, 2007. Paper No. B09.

[240] S. B. Pope. "the evolution of surfaces in turbulence". *International Journal of Engineering Science*, 26:445–469, 1988.

[241] T. Echekki and J. H. Chen. "analysis of the contribution of curvature to premixed flame propagation". *Combustion and Flame*, 118:308–311, 1999.

[242] P. Sripakagorn, G. Kosaly, and J. J. Riley. Investigation of the influence of the reynolds number on extinction and reignition. *Combustion and Flame*, 136:351–363, 2004.

[243] J. S. Brock, J. W. Painter, and D. B. Kothe. Tracer-particle advection: Algorithm components and implementation methods. Technical Report LA-UR-98-5659, Los Alamos National Laboratory, 1998.

[244] H. G. Im and J. H. Chen. Effects of flow transients on the burning velocity of laminar hydrogen/air premixed flames. *Proceedings of the Combustion Institute*, 28:1833–1840, 2000.

[245] S. Shende and A. D. Malony. The TAU parallel performance system. *International Journal of High Performance Computing Applications*, 20(2):287–331, Summer 2006.

[246] John Mellor-Crummey, Nathan Tallent, Michael Fagan, and Jan Odegard. Application performance profiling on the Cray XD1 using HPCToolkit. In *Proceedings of the Cray User's group meeting*, Seattle, WA, 2007. Available as http://www.cs.rice.edu/ johnmc/papers/hpctoolkit-cug-2007.pdf.

[247] Apan Qasem, Guohua Jin, and John Mellor-Crummey. Improving performance with integrated program transformations. Technical Report TR03-419, Department of Computer Science, Rice University, October 2003.

[248] Z. Qin, V. V. Lissianski, H. Yang, W. C. Gardiner, S. G. Davis, and H. Wang. Combustion chemistry of propane: a case study of detailed reaction mechanism optimization. *Proceedings of the Combustion Institute*, 28:1663–1669, 2000.

[249] D. O. Lignell, J. H. Chen, P. J. Smith, T. F. Lu, and C. K. Law. The effect of flame structure on soot formation and transport in turbulent nonpremixed flames using direct numerical simulation. *Combustion and Flame*, 151(1-2):2–28, 2007.

[250] E. R. Hawkes, R. Sankaran, J. C. Sutherland, and J. H. Chen. Scalar mixing in direct numerical simulations of temporally-evolving plane jet flames with detailed co/h_2 kinetics. *Proceedings of the Combustion Institute*, 31:1633–1640, 2007.

[251] Hawkes E. R., Sankaran R., and Chen J. H. Extinction and reignition in direct numerical simulations of co/h_2 temporal plane jet flames. *5^{th} Joint US Combustion Meeting of the Combustion Institute*, page B05, 2007.

[252] Masri A. R. and Bilger R. W. The local structure of turbulent nonpremixed flames near extinction. *Combustion and Flame*, 81:260–276, 1990.

[253] J. C. Sutherland and C. A. Kennedy. Improved boundary conditions for viscous reacting compressible flows. *Journal of Computational Physics*, 191:502–524, 2003.

[254] T. Passot and A. Pouquet. Numerical simulation of compressible homogeneous flows in the turbulent regime. *Journal of Fluid Mechanics*, 181:441–466, 1987.

[255] J. Du and R. L. Axelbaum. The effect of flame structure on soot-particle inception in diffusion flames. *Combustion and Flame*, 100:367–375, 1995.

[256] R. B. Bird and W. E. Stewart and E. N. Lightfoot, Transport Phenomena, Wiley, New York, New York (1960).

[257] W. K. Bushe and H. Steiner, Phys. Fluids, v11, 7, 1896-1906 (1999).

[258] C. M. Cha and H. Pitsch, Combust. Theory Modelling, v6, 425-437 (2002).

[259] J. H. Chen and H. G. Im, Proc. Combust. Inst., v28, 211-218 (2000).

[260] 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).

[261] G. Dixon-Lewis, "Flame Structure and Flame Reaction Kinetics," Proc. Roy. Soc, A., v307, 111-135 (1968).

[262] T. Echekki and J. H. Chen, Combust. Flame, v134, 169-191 (2002).

[263] A. Ern and V. Giovangigli, "Thermal Diffusion Effects in Hydrogen-Air and Methane-Air Flames," Combust. Theory Modelling, v2, 349-372 (1998).

[264] D. Geyer and A. Kempf and A. Dreizler and J. Janicka, Proc. Combust. Inst., v30 (2004) (to appear).

[265] V. Giovangigli, Multicomponent Flow Modeling, Birkhauser, Boston, MA. (1999).

[266] E. R. Hawkes and J.H. Chen, Combust. Flame, v3, 138, 242-258 (2004).

[267] R. Hilbert and F. Tap and H. El-Rabii and D. Thévenin, Prog. Energy Combust. Sci, v30, 61-117 (2004).

[268] , J. O. Hirschfelder and C. F. Curtis and R. B. Bird, The Molecular Theory of Liquids and Gases, Wiley, New York, NY. (1954).

[269] J. C. Hewson and A. Kerstein, Combust. Sci and Tech., v174, 35-66 (2002).

[270] A. Karpetis and R. Barlow, Proc. Combust. Inst., v30 (2004) (to appear).

[271] C. A. Kennedy and M. H. Carpenter and R.M. Lewis, Appl. Num. Math, v14, 4, 397-433 (1994).

[272] A. Kerstein, J. Fluid Mech., v392, 277-334 (1999).

[273] A. Y. Klimenko and R.W. Bilger, Prog. Energy Combust. Sci., v25, 595-687 (1999).

[274] A. Kronenburg and R. W. Bilger, Combust. Sci. Tech., v166, 175-194 (2001).

[275] A. Kronenburg and R. W. Bilger, Combust. Sci. Tech., v166, 195-227 (2001).

[276] A. Kronenburg and A. E. Papoutsakis, Proc. Combust. Inst., v30 (2004) (to appear).

[277] J. Li and Z. Zhao and A. Kazakov and F. L. Dryer, Int. J. Chem. Kinetics (2005) (to appear).

[278] S. Mahalingam and J.H. Chen and L. Vervisch, Combust. Flame, v102, 285-297 (1995).

[279] S. Mitarai and G. Kosĺy and J. J. Riley, Physics of Fluids, v15, 12, 3856-3866 (2003).

[280] S. Mitarai and G. Kosĺy and J. J. Riley, Combust. Flame, v137, 306-319 (2004).

[281] Y. Mizobuchi and S. Tachibana and J. Shinjo and S Ogawa and T. Takeno, Proc. Combust. Inst., v29, 2009-2015 (2002).

[282] Y. Mizobuchi and J. Shinjo and S Ogawa and T. Takeno, Proc. Combust. Inst., v30 (2004) (to appear).

[283] A. Kronenburg and S. Navarro-Martinez, Physics of Fluids (2005) (submitted)

[284] C. Pantano, J. Fluid Mech. (2004) (to appear).

[285] N. Peters, Turbulent Combustion, Cambridge University Press, New York, NY. (2000).

[286] H. Pitsch and C. M. Cha and S. Fedotov, Combust. Theory Modelling, v7, 317-332 (2003).

[287] H. Pitsch and H. Steiner, Physics of Fluids, v12, 10, 2541-2554 (2000).

[288] T. Poinsot and S. Candel and A. Trouvé, Prog. Energy Combust. Sci., v21, 531-576 (1996).

[289] T. J. Poinsot and S. K. Lele, "Boundary Conditions for Direct Simulations of Compressible Viscous Flows," J. Comp. Phys., v101, 1, 104-139 (1992).

[290] S. B. Pope, Prog. Energy Combust. Sci., v11, 119-192 (1985).

[291] R. Sankaran and H. G. Im and E. R. Hawkes and J. H. Chen, Proc. Combust. Inst., v30 (2004) (to appear).

[292] P. Sripakagorn and S. Mitarai and G. Kosly and H. Pitsch, J. Fluid Mech. (2004) (submitted).

[293] J. Sutherland and P. J. Smith and J.H. Chen, Combust. Theory and Modeling (2004) (to appear).

[294] J. Sutherland and C. A. Kennedy, J. Comp. Phys., v191, 502-524 (2004).

[295] L. Vervisch and T. Poinsot, Annu. Rev. Fluid Mech., v30, 655-691 (1998).

[296] J. Warnatz, U. Maas, R. W. Dibble, Combustion , 1^{st} ed. , Springer, Berlin, DE. (1996).

[297] R. A. Yetter and F. L. Dryer and H. Rabitz, Comb. Sci. Tech., v79, 97-128.

[298] S. B. Pope, Prog. Energy Combust. Sci., v11, 119-195 (1984).

[299] S. B. Pope, Proceedings of Sixth International Workshop on the Measurement and Computation of Turbulent Nonpremixed Flames, 2002.

[300] A. N. Karpetis and R. S. Barlow, Proc. Combust. Inst., v30, 665-672 (2005).

[301] C. Pantano, S. Sarkar and F. A. Williams, J. Fluid Mech., v481, 291-328 (2003).