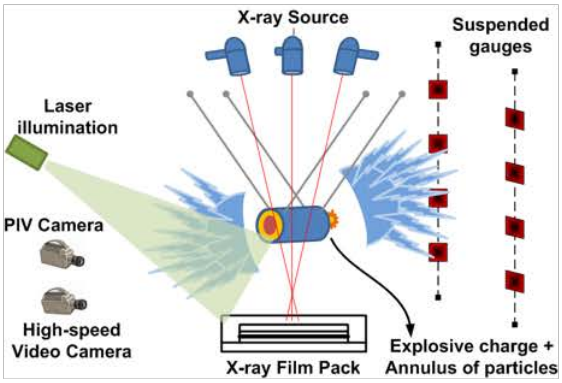
CCMT

CENTER FOR COMPRESSIBLE MULTIPHASE TURBULENCE

Y2 Annual Report

1/27/2015-1/26/2016



UF UNIVERSITY of FLORIDA

NNSA

Table of Contents

1. Introduction

1.1 Background

The University of Florida (UF) established a Center for Compressible Multiphase Turbulence (CCMT) on January 26, 2014 as part of the NNSA's Predictive Science Academic Alliance Program II (PSAAP-II) Single-Discipline Centers (SDC). The intellectual objectives of the Center are threefold: to radically advance the field of compressible multiphase turbulence (CMT) through rigorous first-principle multiscale modeling, to advance very large-scale predictive simulation science on present and near-future platforms, and to advance a co-design strategy that combines exascale emulation with a novel energy-constrained numerical approach. The Center is performing petascale, and working towards exascale, simulations of instabilities, turbulence and mixing in particulate-laden flows under conditions of extreme pressure and temperature to investigate fundamental problems of interest to national technological leadership. Towards this vision we are tackling the following challenges:

Goals of CCMT

- *To radically advance the field of CMT*
- *To advance predictive simulation science on current and near-future computing platforms with uncertainty budget as backbone*
- *To advance a co-design strategy that combines exascale emulation, exascale algorithms, exascale CS*
- *To educate students and postdocs in exascale simulation science and place them at NNSA laboratories*

1) Target an important application that can only be enabled by exascale computing: We are solving a complex multiscale problem at an unprecedented level of physical detail and integration and thereby advance predictive simulation science. CMT poses a grand challenge to our understanding as it combines three complex physics: compressibility, multiphase flow and turbulence. CMT occurs often under extreme conditions of pressure and temperature, and as a result is not easily amenable to high-fidelity experiments and diagnostics. CMT presents a fascinating array of poorly-understood instability, transition, and turbulent processes manifest over a wide range of strongly interacting length and time scales. Current computational approaches involve models and closures that are developed from incomplete understanding, and as a result are largely empirical. Fully validated exascale simulation perhaps is the only path to fundamental breakthroughs that can lead us out of current empiricism.

2) Well-defined problem hierarchy leading to a demonstration problem: A multiscale approach from the microscale to the mesoscale and to the macroscale is being pursued for a systematic integrated investigation of the CMT physics. We have adopted a problem hierarchy that culminates at a signature demonstration problem of explosive dispersal of particles from a well-characterized initial condition, which fully exercises all the key complex processes of CMT. We pursue a coupling strategy where (i) fully resolved microscale simulations will lead to reduced order descriptions (interphase coupling models) to be employed at the mesoscale and (ii) partially resolved mesoscale simulations will lead to reduced order descriptions (multiphase large eddy

simulation closures) to be employed at the macroscale. This will allow computational efficiency and high degree of parallelism at all levels of the hierarchy.

3) Simulation and experiment roadmaps for rigorous validation: We focus on integrated system-scale simulations of the demonstration problem from the outset using existing integrated code capabilities. Simultaneously, we also perform petascale simulations at the micro and mesoscales. Improvements to micro-to-meso and meso-to-macro coupling models will be systematically and periodically incorporated at the appropriate higher level. A layered systems engineering approach is used to organize and integrate physical subsystems with numerical, software and service components, to achieve progressively improved operational capability for system-scale simulations. We have developed a detailed simulation and experiment roadmap which allow rigorous step-by-step validation at each step of the problem hierarchy.

4) Develop novel uncertainty quantification (UQ) approaches for CMT: Detailed measurements from carefully chosen existing and planned experiments at the Air Force Research Laboratory Munitions Directorate (AFRL-RW), Sandia Multiphase Shock Tube facility and Los Alamos Center of Mixing under Extreme Conditions (CoMuEX) are used for rigorous quantification of uncertainties from the micro/mesoscales to the macroscale. We are engaged in vigorous uncertainty reduction through better characterization and instrumentation, rigorous calibration of the models, and improved numerical resolution. Simultaneous simulations and experiments at the micro, meso and macroscales of the problem hierarchy will allow us to both propagate up uncertainty to higher scales, and to reduce uncertainty through iterative improvements at the lower scales. A particularly difficult aspect of CMT is that it is characterized by extreme events that are localized in space and time. A key innovation is the development of novel techniques for accurate characterization of probability tails in the uncertainty quantification of such rare but critical events.

5) Demonstrate integrated performance on current/near-future architectures: Modern many-core architectures (such as Intel MIC), that provide high raw gigaflops, have deep memory hierarchies and low overhead threading capabilities. We exploit these capabilities to optimally utilize both computational and energy resources. In particular, we will tackle load balance and performance challenges in terms of data and work decomposition for the CMT code framework. Different parallelization schemes will be considered for effectively implementing simulations at the microscale, mesoscale, and system-scale, especially for heterogeneous resources.

6) Develop methods for predicting performance on a variety of exascale architectures: While many exascale trends seem clear, there are far too many permutations in the design space to select one a priori. We leverage the unique Novo-G facility at the NSF-supported UF Center for High-Performance Reconfigurable Computing (CHREC) to emulate and evaluate a series of candidate exascale architectures. We are developing an unprecedented capability to behaviorally prototype in software and hardware a variety of promising (as defined by leading exascale initiatives) forms of next-generation exascale (i) device and node designs at the micro-level and (ii) communication and system architectures at the macro-level. We are conducting experiments with CMT-bone

kernels, miniapps and skeleton-apps to evaluate promising architectures in terms of performance, energy, temperature, reliability, and scalability. Modeling, simulation, and estimation tools (e.g., those supported within the Sandia's Structural Simulation Toolkit (SST)) are being leveraged with our behavioral simulations and emulations.

7) Solutions for energy efficiency and thermal management: We are developing a framework for multi-element and multi-objective optimization that will simultaneously minimize energy and maximize performance. We exploit the data and task parallelisms within CMT application and its UQ implementation to develop innovative low complexity static and dynamic algorithms for scheduling, while considering important factors such as thermal constraints and leakage currents.

1.2 Demonstration Problem

We aim at solving a problem of Compressible Multiphase Turbulence (CMT) at an unprecedented level of physical detail and thereby advance predictive simulation science. The overarching demonstration problem consists of a cylindrical core of simple explosive pellet of about 10 grams will be surrounded by a cylindrical very-thin-walled glass jacket of larger diameter. The annular region between the pellet and the jacket will be filled with mono or polydisperse metal powder of spherical shape. The shape and amount of the explosive charge and the size distribution of the metal powder and its material (aluminum, steel, tungsten, etc.) are parameters that will be varied. The charge will be hung from a test fixture so that the effect of the ground and the surrounding structures will be eliminated during the initial phase of explosion and dispersion. The orientation of the test setup will be such that the resulting explosive dispersal of particles and the gas field can be highly accurately measured. The following features makes this problem a very good choice for demonstration: (i) the explosive dispersal exercises all the major CMT physics, (ii) the extreme conditions makes this a demanding test for predictive capability, (iii) this problem requires exascale for true predictive capability, and (iv) we have already performed similar experiments

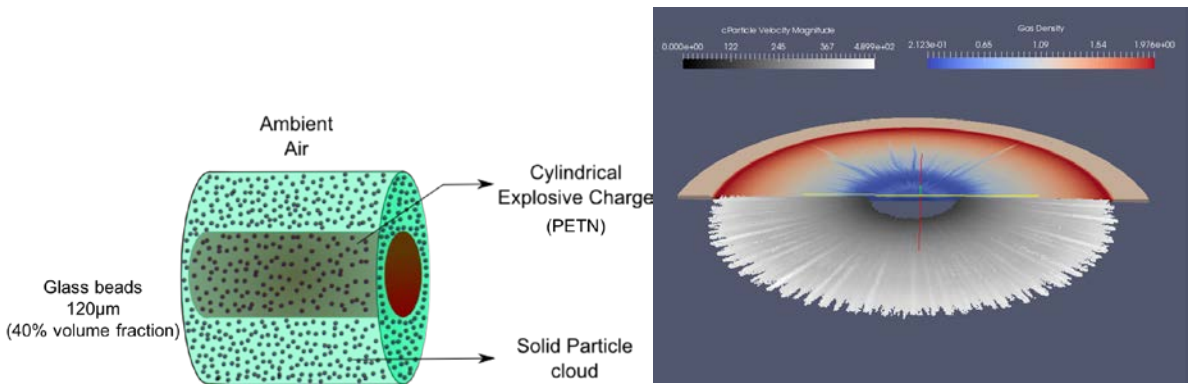


Figure 1.1. Left: Schematic of the three-dimensional computational domain for the explosive dispersal of particles simulations. Not drawn to scale. Right: Current three-dimensional hero run: 30 million computational cells, 5 million computational particles, computational domain extended to about 4m in radius. Time shown, t=1.46ms.

and validation-quality measurements. The explosive dispersal of solid particles problem displayed in Figure 1.1 and described by Frost *et al.* (Phys. Fluids, 24(9), 2012) was chosen for the initial phase of our research activities.

1.3 Simulation Roadmap

The center is focused on integrated system-scale simulations of the demonstration problem from the outset using existing integrated-code capabilities. Figure 1.2 shows the roadmap of the proposed sequence of simulations. The following important considerations was used in constructing the roadmap: (i) Along with system-level simulations of the demonstration problem, we will perform increasingly more complex simulations at the micro and mesoscales. Based on these simulations, improvements will be made to micro-to-meso and meso-to-macro coupling

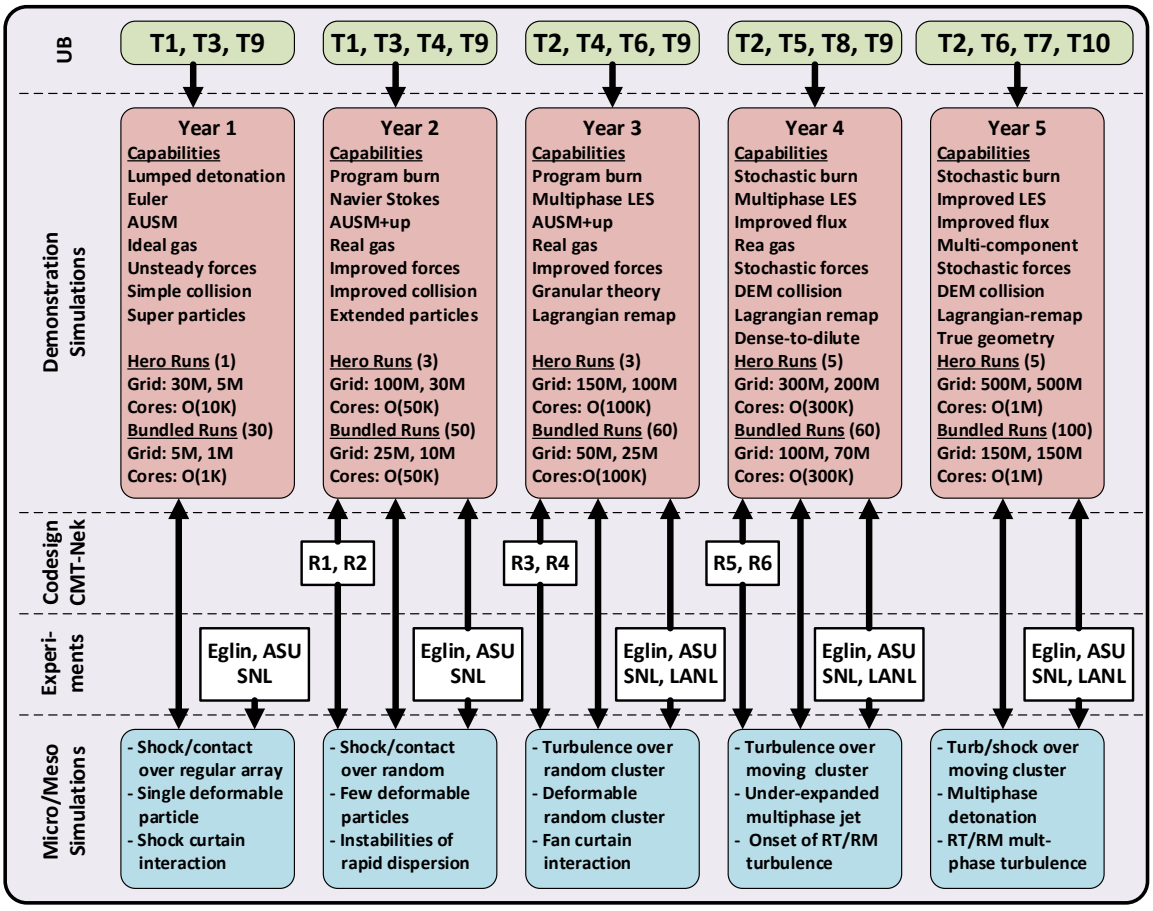


Figure 1.2. A roadmap showing the sequence of progressively more complex simulations to be performed over the five years of the center. Micro and mesoscale simulations will be performed along with macroscale simulations of the demonstration problem. Both *hero runs* and *bundled runs* will be performed. Bundled runs will be combinations of varying parameters and ensemble of realizations. This plan is based on future petascale platforms that are expected to be available through NNSA. Concurrent simulations will be accomplished using DAKOTA.

models. (ii) To take maximum advantage of validation experiments, large numbers of simulations will be required for optimal calibration. We are using surrogate models to allow us to solve the multi-level optimization problem associated with selecting the physical constants that give the best match with the numerical model. (iii) Variations of the key control parameters (particle size, particle material, shock strength, etc.) will be guided by simulations that identify which combinations of parameters will elicit different modes of instability. (iv) Statistical variability will be explored through an ensemble of realizations under nominally identical conditions. (v) Simulations are currently being carried out concurrently as *bundled runs* using the DAKOTA toolkit. (vi) We anticipate increasingly larger petascale computational platforms to be available at the NNSA labs. (vii) We have and will continue to perform selective *hero runs* at super-high resolution to help quantify discretization errors to help assess the accuracy of the estimated uncertainties. (viii) UQ is being used to guide the selections of quantities to be measured with preference to those with low uncertainty, so as to avoid empty validation based on large error bars.

The Year-1 simulations of the demonstration problem employ simplified physics model: (i) a lumped detonation model, (ii) the single-phase AUSM+ flux scheme for the Euler gas equations with idea gas equations of state, (iii) the actual particles are approximated with computational super particles, (iv) gas-particle coupling is through point-particle models of quasi-steady and unsteady forces and heat transfer, and (v) particle-particle collisions are accounted using a simplified collision model. The corresponding hero and bundled runs represent our Year-1 starting point. The above roadmap shown in Figure 1.2 lays out year-by-year progression of more detailed simulations that incorporate additional physics through new and improved models. Furthermore, each year we plan to perform larger and larger hero runs as well as large array of bundles macroscale simulations for uncertainty quantification.

The simulation roadmap is driven from the top by Uncertainty Budget (UB). A detailed phenomenon identification and ranking analysis of the demonstration problem has identified 11 key sources of errors and uncertainties which are briefly listed below:

- T1: detonation process modeling
- T2: Multiphase turbulence modeling
- T3: Real gas thermodynamic and transport properties
- T4: Inter-particle collision modeling
- T5: Particle compaction modeling (during early stages of detonation/shock propagation)
- T6: Point particle modeling of gas-particle momentum (force) exchange
- T7: Point particle modeling of gas-particle thermal (heat-transfer) exchange
- T8: Particle deformation, sintering and break-up physics
- T9: Discretization (numerical) errors
- T10: Errors from geometric approximation (geometric differences in the details of experiments and simulations)
- T11: Experimental uncertainties and measurement errors

The key activity of UB effort will be to quantity the uncertainty in the zeroth and first order prediction metrics. The zeroth order prediction metrics of the demonstration problem are:

- The blast wave location as a function of time
- The average particle front and tail locations as a function of time
- The number of large-scale instabilities of the particulate front

The first order prediction metrics go beyond the zeroth order metrics and the details of the demonstration will be quantified with the following first order metrics:

- Time evolution of the pressure at selected points within 5% error
- Time evolution of the thermal load at selected points within 20% error
- Time evolution of average particle concentration within 15% error
- Evolution of particle front thickness due to instability and turbulent mixing within 10% error
- RMS turbulent velocity and pressure fluctuations at the particle front within 15% error,
- Time evolution of local particle size distribution within 15% error
- Multiphase turbulent spectra and correlation length scales within 20% error.

An important component of the yearly UB effort is to quantify contribution from the above 11 sources of errors and uncertainties to each of the prediction metrics. This quantification will allow us to focus on error/uncertainty reduction. Thus each year we will focus on uncertainty reduction and quantification through certain modeling and simulation activities. These are the UB drivers for the proposed roadmap and they are presented at the top row of Figure 1.2.

Figure 1.2 also presents the yearly releases of CMT-nek, the new code being co-designed through an integration of exascale higher-order algorithm with exascale emulation/ simulation. Also indicated are yearly coordination with the micro, meso and macroscale experiments to be performed at Eglin Air Force Base, Arizona State University (ASU), Sandia National Laboratory (SNL) multiphase shock tube facility and Los Alamos National Laboratory (LANL) Center of Mixing Under Extreme Conditions. The macroscale simulation road map will also be supported by the yearly progression of mico and mesoscale simulations, which is also indicated in Figure 1.2.

1.4 Integration

The Center recognizes the critical importance of tight integration for the success of the center. The center will be organized in terms of tasks and cross-cutting teams, rather than in terms of faculty and their research groups. The physics-based tasks are continuous and particulates phase modeling and simulation. In addition we have exascale (EX), computer sciences (CS) and uncertainty quantification (UQ) as the cross-cutting tasks that will interface and integrate the physics-based tasks. By ensuring faculty, research scientists, and postdocs contribute to multiple physics and/or cross-cutting tasks, we will achieve tight integration. This matrix organization, depicted in Figures 1.3 and 1.4, tears down discipline and departmental boundaries and allows close interaction. In addition, significant effort has gone into integrating the various disciplines.

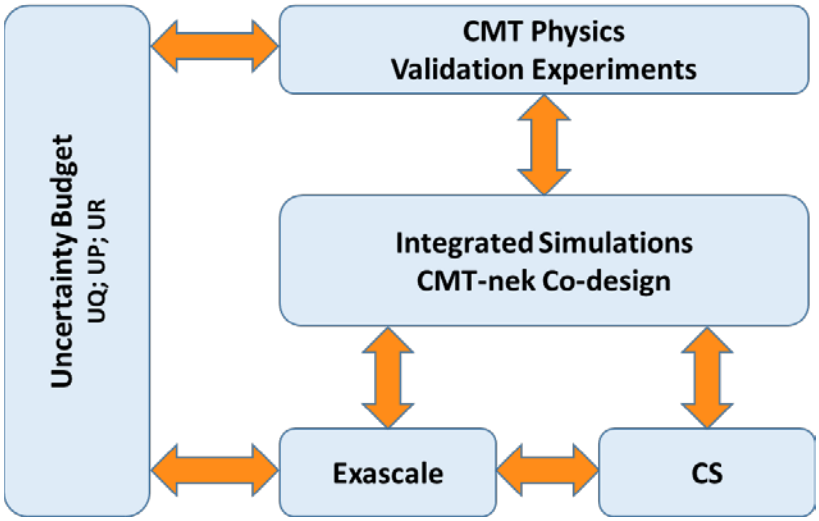


Figure 1.3. Integration – how the different pieces fit together. Here, UB drives the CMT physics, validation experiments, and exascale behavioral emulation/simulation. CS and exascale feeds into the CMT-nek codesign strategy.

Hour time slots	Exascale	CMT-nek	CS	Micro	Macro	UQ	Exp
Exascale	X	X	X			X	
CMT-nek	X	X	X	X	X		
CS	X	X	X				
Micro		X		X	X	X	
Macro		X		X	X	X	X
UQ	X			X	X	X	X

Figure 1.4. Management – tasks and teams. Teams include students, staff, and faculty. The Center is organized by physics-based tasks and cross-cutting teams, rather than by faculty and their research groups. All staff and large number of graduate students located on 2nd floor of PERC. All meetings held in PERC. Weekly interactions (black); Regular interactions (red).

The intellectual integration of the different simulation and experimental talks, across the three different scales (micro, meso and macro) is shown in Figure 1.5. Uncertainty quantification, propagation and reduction along the ten sources of errors/uncertainties (T1 to T10) forms the framework that connects and drives the different simulation and experimental activities of the center. The hierarchical flow of error/uncertainty information to the macroscale is shown.

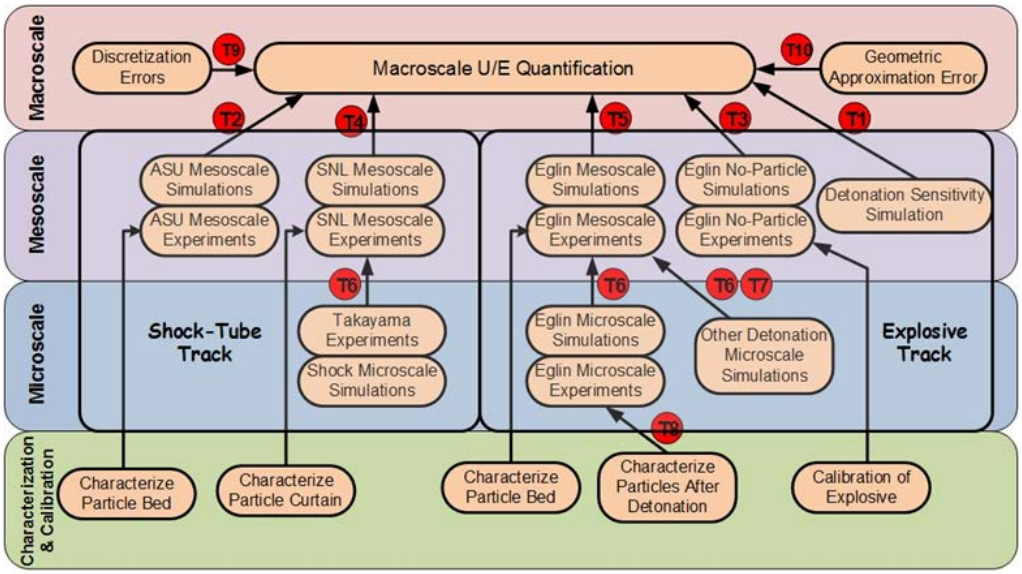


Figure 1.5. Uncertainty Quantification and Propagation Plan.

At the *microscale* the motion and thermal evolution of particles depends on the flow around them. In return, the particles modify the local flow by the formation of momentum and thermal wakes. Particle structures (chains and clusters) spontaneously form due to wake-wake, particle-wake and particle-particle interactions. At the *mesoscale*, due to inertial interaction with turbulence, particles preferentially accumulate. Also, flow instabilities can lead to large-scale structures in particle distribution. These nonuniformities have profound influence on their collective back influence on the flow. At the *macroscale* (or *system-scale*) the geometric details of the setup influence the coupling between the particles and expanding gas. Important aspects of the multiscale coupling strategy we are pursuing includes: (i) microscale-informed reduced-order descriptions (point-particle coupling models) to be employed at the mesoscale and (ii) mesoscale-informed reduced-order descriptions (multiphase LES models) to be employed at the macroscale. With this strategy, the predictive capability at the system-scale can be thoroughly validated and uncertainty rigorously quantified as illustrated in Figure 1.5.

Note that the multiscale coupling strategy and the overall uncertainty quantification plan includes both a shock-tube track and an explosive track. We have been working with the Experimental Teams at the various locations and have discussed in detail the type of characterization, inputs, and output from the experiments for a meaningful UB approach.

Finally, Figure 1.6 shows the timeline for performing the different tasks. These tasks T1-T11 were previously described.

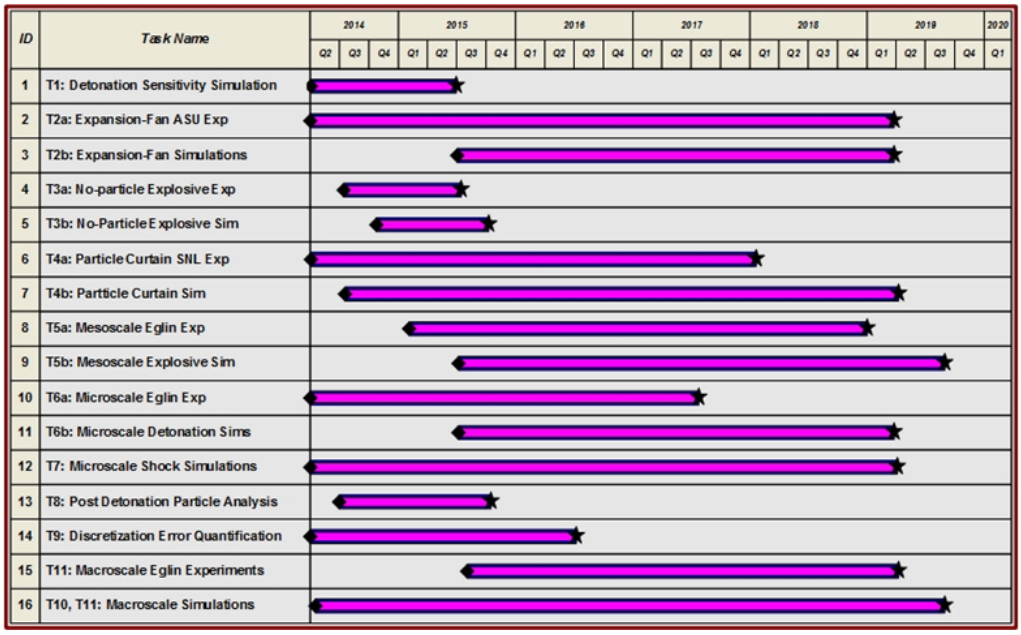


Figure 1.6. Uncertainty Quantification Task Timeline.

2. Macroscale Team

2.1 Overview

For the Macro/Mesoscale Simulation Team (MMST), year two of CCMT was synonymous of substantial progress on the physics used in our code, and of transition to consistent production runs. The goal of MMST remained unchanged. Namely, we aim at running a problem of explosive dispersal of particles at an unprecedented level of details. See Fig. 2.1 for a depiction of the problem.

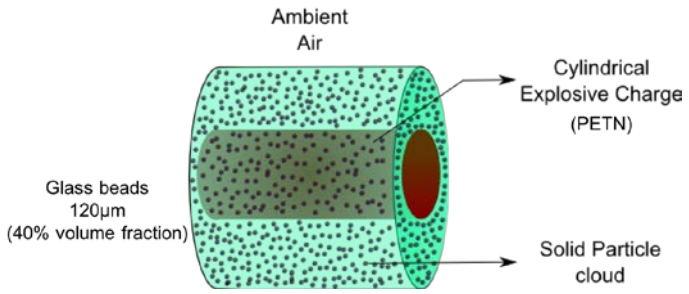


Figure 2.1. Schematic of the three-dimensional computational domain for the explosive dispersal of particles simulations. Not drawn to scale.

Our strategy has been the following: 1) keep scaling up the simulation size in terms of resolution and core count to be able to capture detailed physics; 2) run simulations to physical time closer to the reference experiment – nearly 10 times longer than during year 1 – to be able to provide meaningful comparisons of the prediction metrics; 3) improve the least-up-to-date models of our code to provide higher quality solutions.

2.2 Code Scaling

Pushing the limit in resolution and core count for our simulation of the signature problem is a yearly objective for the team. The knowledge accumulated during year 1 indicated that a very targeted effort could allow a substantial increase in scale of our hero run.

In particular, the generation of very large grids was limited by Rocflu's partitioner. The partitioner is required to allow the CFD code to spread the computational domain among numerous processors. In the version of Rocflu used in year 1, this partitioner was serial, thus causing unreasonable amount of time and substantial memory requirement for grid sizes targeted for our hero runs. A focus team within the MMST was created to parallelize the partitioner, allowing several processors to work on different regions of the computational domain at the same time. Their work resulted in a massive speed up of the partitioning, which, in turn, allowed for using increased computational domain resolution. The large memory requirement remains to be addressed, but does not constitute a roadblock to our progress for now.

It followed that before the end of year two, we have been able to spectacularly fulfill our objective of scaling up Rocflu relative to year 1. Indeed, we went from a hero run of 30 million computational cells and 5 million computational particles on 512 cores at the end of year 1, to 50 million computational cells and 10 million computational particles on 8192 cores at the end of year 2. This simulation was run on Vulcan at LLNL, and multiple restarts were used to reach about 10μs of physical time. Starting from an initial value of 5%, the particle volume fraction reaches

the close packing limit very early during the compaction phase of the explosive dispersal. Our post-run analysis showed two key results: i) the collision model used in our code needs to be improved to be able to handle a larger density of computational particles; ii) an optimization needs to be accomplished on our code to better take advantage of the large core count.

The work addressing of both these drawbacks has already begun. A team effort for improving the collision model has been planned, and given high priority. The optimization effort is, on the other hand, a continuous effort. After demonstrating the code capability to run 16K cores on a gas only problem in year 1, we have now made proof in year 2 that the full demonstration problem (gas + particles) runs with 8K cores. The effort in improving the efficiency of the code becomes then crucial to justify the next round of increase in core count and grid resolution.

2.3 Progress on the Demonstration Problem

Our effort in pushing the limits on our Hero run in year two has been motivated by the will to have meaningful comparisons of the prediction metrics. For memory, four prediction metrics have been defined for judging simply and effectively of the predictive capability of our simulation code: 1) the blast wave location, 2) the outer-front location the particle cloud, 3) the number of aerodynamically stable particle jets at late time, and 4) the size of the aerodynamically stable particle jets. Because of the limited physical size chosen for our computational domain in year 1,

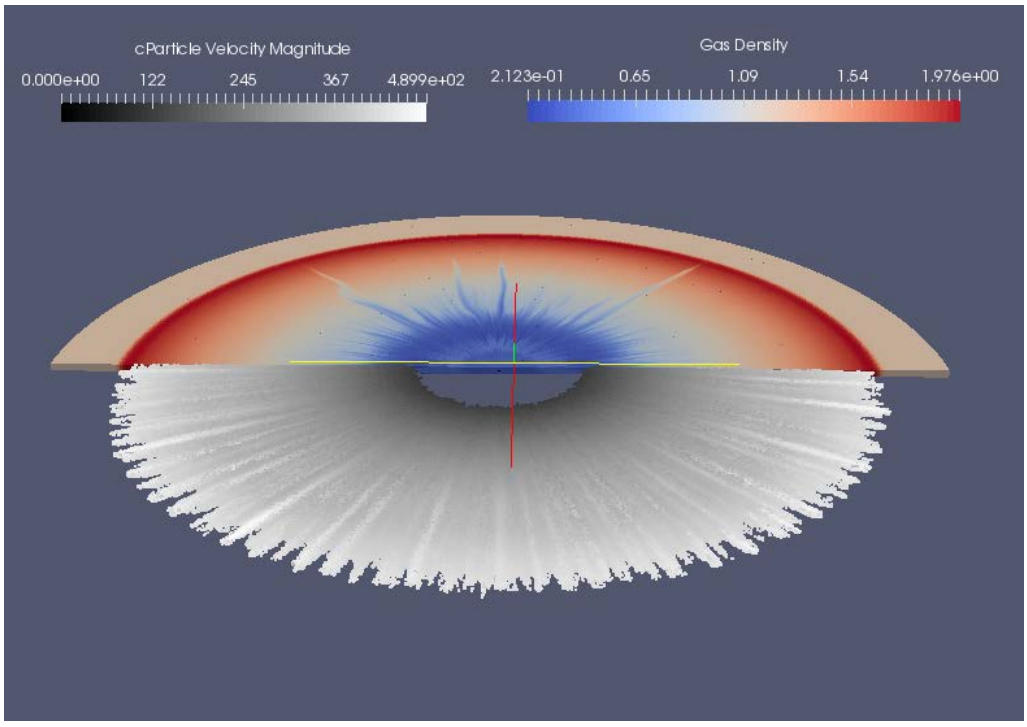


Figure 2.2. Current three-dimensional hero run: 30 million computational cells, 5 million computational particles, computational domain extended to about 4m in radius. Time shown, t=1.46ms.

the duration of our solution to the demonstration problem could not exceed 200μs. Balancing the domain size, the grid resolution, the computational particle count, and the performance of the code, we have designed a computational domain sufficient to capture up to 2ms after the initial detonation of the energetic material, see Fig. 2.2. This simulation is most efficient with 4K cores, and has reached 75% completion as this report is written. Comparisons of metrics 1) and 2) between our simulation and the experiment show that our current hero run predicts a faster blast wave, but a slower particle cloud expansion; see Fig. 2.3. Comparisons of metrics 3) and 4) may be possible by the completion of the run if aerodynamically stable jets are observed/formed.

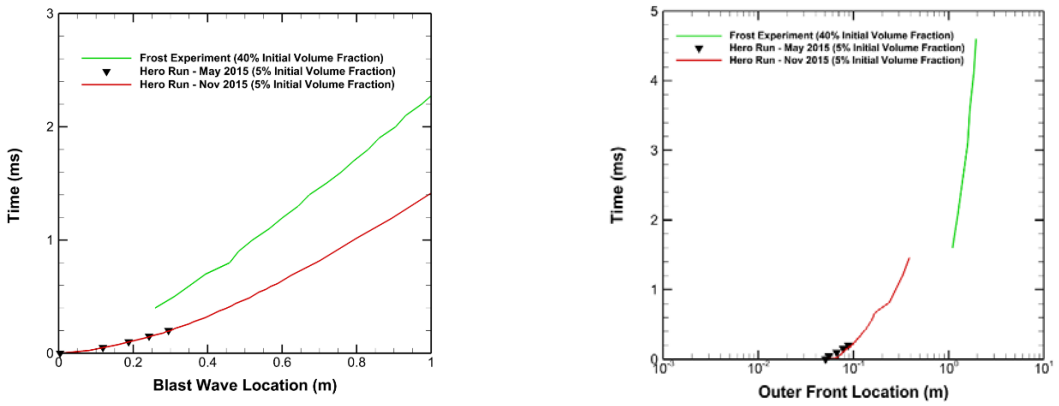


Figure 2.3. Left – Blast wave location as a function of time (PM-1). Right – Particle cloud outer front location as a function of time (PM-2). The results depicted in red and labelled "Hero-Run – Nov 2015" are extracted from the current Hero Run. The current run is a continuation of the run for which results were presented at the November 2015 TST review meeting. Its current physical time is t = 1.465ms.

Despite our already impressive grid size (50 million computational cells), our hero run remains under resolved to capture all the minute physics of gas and particles. Nevertheless, the lack of grid resolution taken aside, analyses made on the demonstration problem during year two clearly confirmed a need for the improvement in physical models. The work addressing this challenge has begun during the last quarter of year 2, and some important milestones have already been accomplished.

2.4 Physical Models Improvement

The implementation of a real gas equation of state and of a more appropriate flux scheme were important goals we set and achieved during year 2.

Test runs during year 1 indicated that the temperature of a PETN explosion predicted in our simulation was not agreeing with published experimental values, when using the perfect gas equation of state. The implementation of a real gas equation of state into the code then became a

priority. The Jones-Wilkins-Lee (JWL) Equation of State (EoS) for explosives was implemented into the code for mixtures of ambient air and gaseous detonation products using an iterative method. The method would take either mixture density and pressure or mixture density and energy (along with the product mass fraction) and go one of three ways: (1) Use the Ideal Gas EoS if no product is present, (2) Use the JWL EoS if only product is present or (3) Use a Broyden iterative solver for a mixed cell until pressure and temperature equilibrium is satisfied in the cell.

To help make this procedure faster, work was also done on creating a multi-fidelity surrogate model with help from the UQ team that would take the input parameters mentioned before and output the equilibrium state of mixed cells without the need for iteration. A sample plot of this surrogate at a mass fraction of 0.5 is shown in Fig. 2.4. An added potential benefit of this surrogate is the capability of using it to determine if the inputs given to the JWL function from the code are in a physically possible range (so, this would be a check against model/numerical errors in the code). This would be possible through the use of a convex hull that was created for the input space given by (mixture density, mixture energy, product mass fraction) when the surrogate model was developed. This convex hull was created during a back and forth process with the UQ team when they were requesting data from the iterative solver to create the surrogate. Once we started running

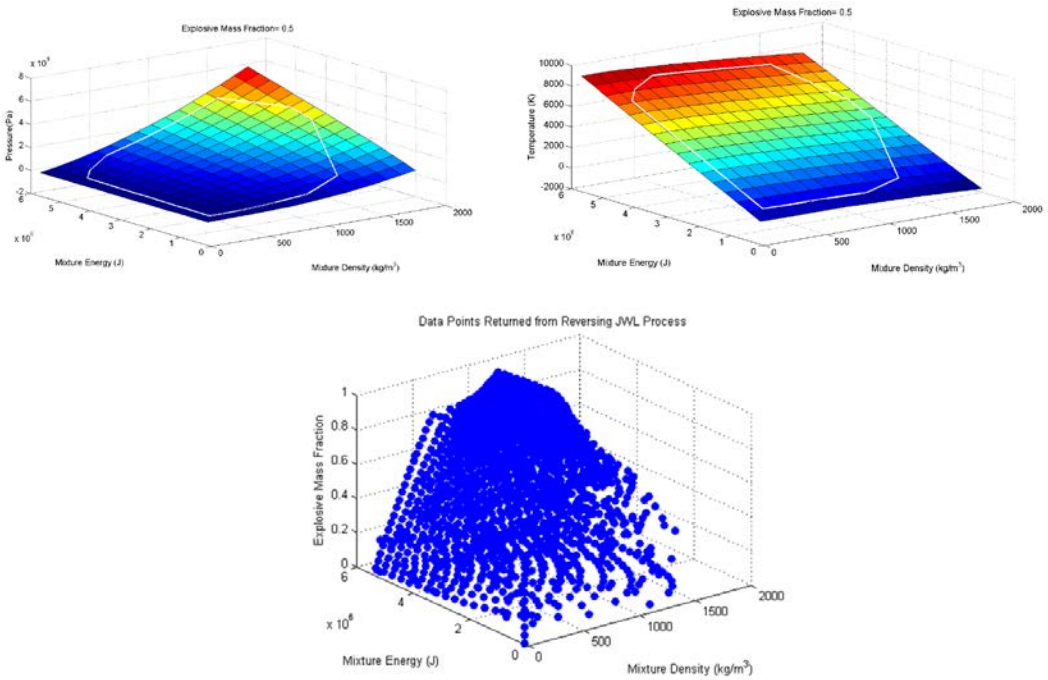


Figure 2.4. Top - Slices of the multi-fidelity surrogate for pressure (left) and temperature (right) at an explosive mass fraction of 0.5 for the ranges of mixture energies and densities that are encountered during the simulation. The white curve represents the boundary between the physically possible and physically impossible points in the domain. Bottom - Scatterplot of all of the test points in the input variable domain which yield a physically possible solution when put into the surrogate model. This is also referred to as the convex hull of the domain.

into points in the input domain that did not converge in the iterative method, they had the idea of creating a boundary between these points and the points that did converge. An image of the convex hull is also shown in Fig. 2.4.

Next, an Eulerian-Lagrangian formation of the multiphase AUSM+up scheme has been implemented. The AUSM+up flux scheme was originally designed to improve stability for low Mach number single phase flows and was later extended for use in multiphase Eulerian-Eulerian flow simulation. UF CCMT team members were able to extend the Eulerian-Eulerian formulation of AUSM+up to an Eulerian-Lagrangian formation to accommodate the multiphase Eulerian-Lagrangian simulations of interest. A thorough validation of the implementation of the Eulerian-Lagrangian AUSM+up flux scheme has been completed, and all newly submitted run are now using it.

Lastly, the collision model is under thorough investigation to identify options for improvement at the time this report is written.

2.5 Collaborative Effort

The Macroscale team has been working in close collaboration with the Uncertainty Budget (UB) team. As seen in the previous section, a significant amount of effort has been invested in the development of a multi-fidelity surrogate model for the compute of real gas equation of state solutions. This approach is unique, and may constitute a highly efficient and reasonably accurate method for multi-species mixing under real gas conditions. Testing and evaluation of this approach will be led by the UB and MMST at the beginning of year 3.

Also, the UB team, the experimentalist in SNL, and the Macroscale team have worked jointly on the multiscale uncertainty budget and error reduction using the (mesoscale) particle curtain problem. Analyses and comparisons of 1D simulations, 2D simulations and experimental results have led to new shock tube data collection with new instrumentation (X-Ray). New analyses are underway to better identify areas that offer the best opportunity for uncertainty reduction.

Finally, during year two, the Macroscale team has continued its regular interaction with the experimental team at Eglin Air Force base, and with the experimental team at Arizona State University. The topic of discussion has predominantly been concerned with the design and instrumentation of experiments.

2.6 Third Year Plans

Regarding improvement to Rocflu, the emphasis in year three will be performance. The MMST will work at drastically speeding-up the code in such a way that it takes full advantage of the larger core count that can now be achieved (>8K). Next, the team will be integrating full co-processing capabilities to Rocflu with the introduction of Catalyst, as it has been postponed during the course of year 2.

Concerning the physics models, the team will be focusing on implementing a new Lagrangian-based volume fraction computation, and a much needed collision/compaction model. Research efforts on a multiphase LES model will continue but will be lower priority.

Finally, the collaboration with the Microscale team will strengthen in year three as they are completing new particle force models. Integration of these models and testing shall take place during the course of year three. Similarly, the collaboration with the UB team will continue on the mesoscale particle curtain problem, and on finalizing the multi-fidelity surrogate approach for compute of real gas equation of state solutions. Collaborations with the experimental teams shall continue and intensify toward the end of year three when results of the latest experiments will be available.

3. Microscale Team

3.1 Overview

The purpose of the microscale simulations is to develop point-particle (PP) and other related models that are needed for the mesoscale and macroscale simulations of explosive dispersal of particles. These particle models are required to compute (a) instantaneous aerodynamic force on the particle and (b) instantaneous net heat transfer between the particle and the surrounding. The commonly employed standard models for drag and heat transfer have been derived under simplifying assumptions. The objective here is to develop extended models in finite Mach, Reynolds, and volume fraction regimes – to be used in mesoscale and macroscale simulations. A strategy for a sequence of microscale simulations has been devised that will allow systematic development of the hybrid surrogate models that are applicable at conditions representative of the explosive dispersal application. The ongoing microscale simulations seek to examine particle force dependence on Mach number and volume fraction.

Over the past year, the microscale simulation team has been working on direct numerical simulations (DNS) of shocks interacting with multiple particles. The purpose of these simulations is: (a) to study the transient force and the force fluctuations experienced by the particles and (b) to apply this knowledge and use the results from simulations for developing the point-particle models. The shock-particle interaction simulations include structured arrays of parrticles and random particle distributions, as discussed below. Also, a grid resolution study is being conducted for multiple particle simulations in collaboration with the Uncertainty Quantification (UQ) team at the center. Multiple simulations are being performed by varying the mesh resolution on the surface of the particle as well as the global mesh in the computational domain. Richardson extrapolation is being employed to estimate error percentage for each of these simulations. The conclusions of this grid resolution study will br reported in the future.

3.2 Year 2 Research

For the microscale simulations, two compressible flow solvers are being employed. *Rocflu* is a parallel finite volume solver (developed under the ASAP program) for unstructured, body-fitted meshes. It can solve inviscid or viscous compressible flows and is being used for microscale simulations for non-deforming particles to study shock-particle and contact-particle interactions including effects of Reynolds number, Mach number, volume fraction. *RocSDT* is a compressible code based on level set methodology on Cartesian grids and can handle shock-particle interactions, multi-material interfaces, material deformation, general equations of state, and chemistry. It complements *Rocflu* in two ways: (a) it allows particle deformation, and (b) it allows more flexibility for arbitrary/random particle clusters. These codes are then applied to the following problems of a shock propagating over a structured arrays of particles and a random bed of particles.

Shock Interaction with Structured Arrays of Particles

Shock interaction with single particle

This is the case with zero volume fraction in which a shock travels over a single isolated spherical particle, which the particle is assumed to be non-reactive and rigid. We are interested in the short time force history experienced by the particle due to the impact of the shock and for this case we can assume the particle to be stationary. Flow is assumed to be inviscid for all the cases. For these

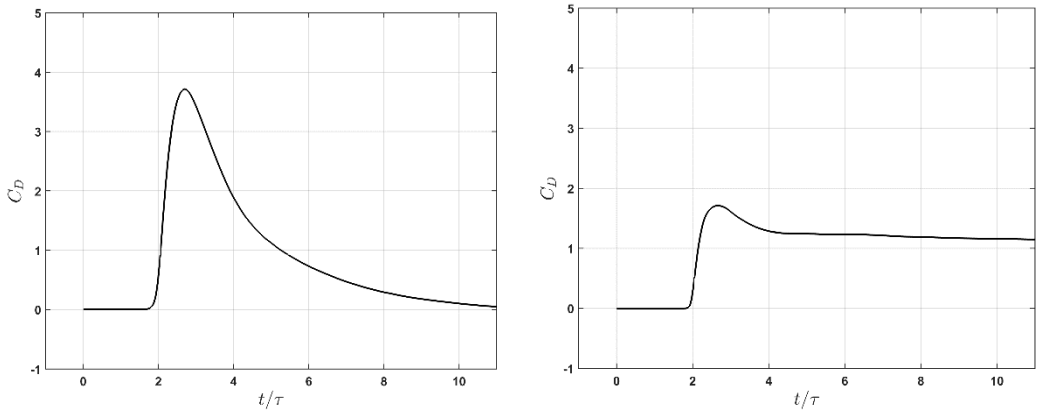
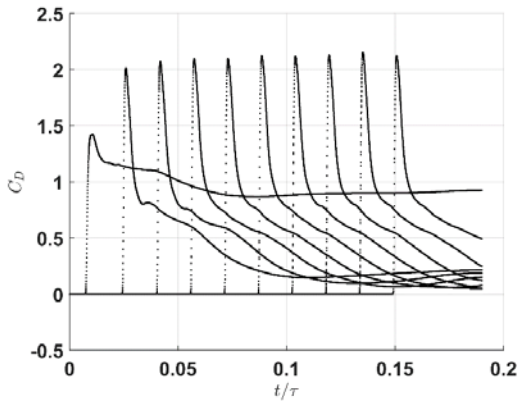


Figure 3.1. Plot of non-dimensional drag (C_D) against non-dimensional time (a) for M_s=1.22 (b) and M_s=6.0.

cases force history or the drag coefficient experienced by the particle is just a function of shock Mach number. We have carried out multiple simulations, where shock Mach number (M_s) was varied to establish the effect of the same on the force history. For M_s=1.5 the post shock Mach number is less than critical (0.6), whereas for M_s=3.0 it is more than the critical Mach number. Hence, for M_s=1.5 drag on the particle goes to zero after about 20 time scales, but for M_s=3.0 a bow shock is formed in front of the spherical particle which results in non-zero post shock drag experienced by the particle. Figure 3.1 shows streamwise drag as a function of time for two different Mach numbers.

Shock interaction with one dimensional structured array of particles

Another limiting case of the shock interaction with randomly distributed particles is the shock interaction with the one-dimensional structured (horizontal) array of particles. For shock Mach number equal to 6.0, we can observe from Figure 3.2(a) that the drag increases for every successive particle in the array. There is a big jump in the drag for the second particle as compared to the first particle, and the drag levels off after the fifth particle. For this case, as seen in Figure 3.2(b), bow shocks are formed in front of each particle. There is a complex wave interaction in between the particles which results in an increase in peak drag for successive particles as compared to the previous particle. This increase in peak drag is a function of inter-particle spacing and shock Mach number. These simulations help us in understanding the dynamics of shock interaction with a structured bed of particles and to get an estimate for asymptotic increase in drag for a horizontal layer of particles.



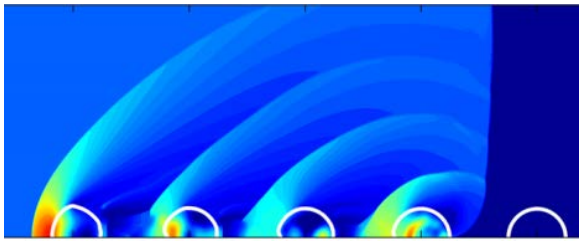


Figure 3.2. (a) Plot of non-dimensional drag coefficient (C_D) against non-dimensional shock time scale (t/τ). (b) Density contour plot. Shock Mach number (M_s) = 6.0.

Shock interaction with one layer of transverse array of nine particles

We have also conducted simulations of shock interacting with one layer of transverse array of nine particles. Figure 3.3 shows the three-dimensional computational domain and results are presented in Figure 3.4. For these simulations we varied the particle spacing L/d_p as 4, 3, 2, 1.5 and 1.2 and compared the drag on the particles against the drag on a single isolated particle. We performed these simulations for M_s = 3.0 and 6.0. From these simulations we can understand the effect of neighboring particles on the drag.

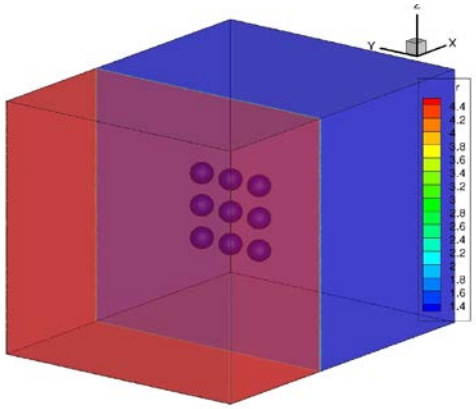
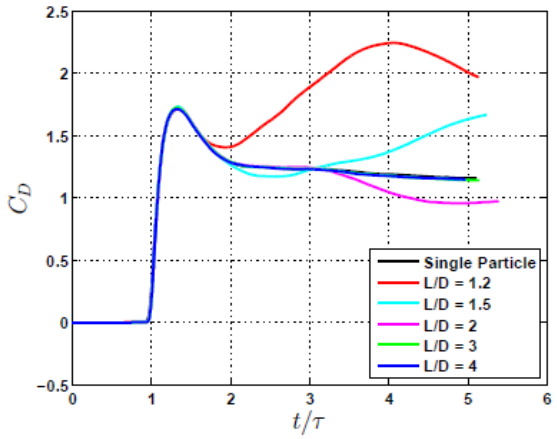


Figure 3.3. Three-dimensional domain for shock interaction with one layer of transverse array of particles; M_s=3.0 and 6.0.

Shock interaction with FCC array of particles

To study the effect of the volume fraction along with the effect of the shock Mach number, we have carried out simulations of shock interaction with particles arranged in a face centered cubic (FCC) unit cell for different shock Mach numbers and volume fractions. Table 3.1 shows the matrix of simulations and Figure 3.5 shows the three-dimensional domain. Results are presented in Figures 3.6 and 3.7. These simulations will help us understand the effect of volume fraction and shock Mach number on the drag experienced by the particles due to a shock passing over them in the presence of mulitple particles.



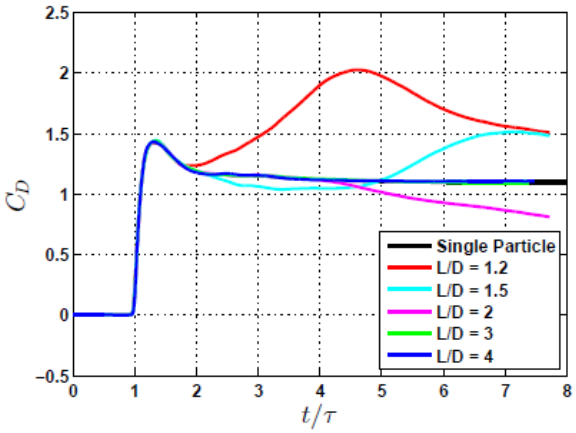


Figure 3.4. Plot of non-dimensional drag (C_d) against non-dimensional time (a) for M_s=3.0 (b) and M_s=6.0.

Table 3.1: Array of simulations for shock interacting with FCC arrangement of particles.				
	$\phi = 10\%$	$\phi = 20\%$	$\phi = 30\%$	$\phi = 40\%$
$M_s = 1.5$	RUN1	RUN5	RUN9	RUN13
$M_s = 2.0$	RUN2	RUN6	RUN10	RUN14
$M_s = 3.0$	RUN3	RUN7	RUN11	RUN15
$M_s = 6.0$	RUN4	RUN8	RUN12	RUN16

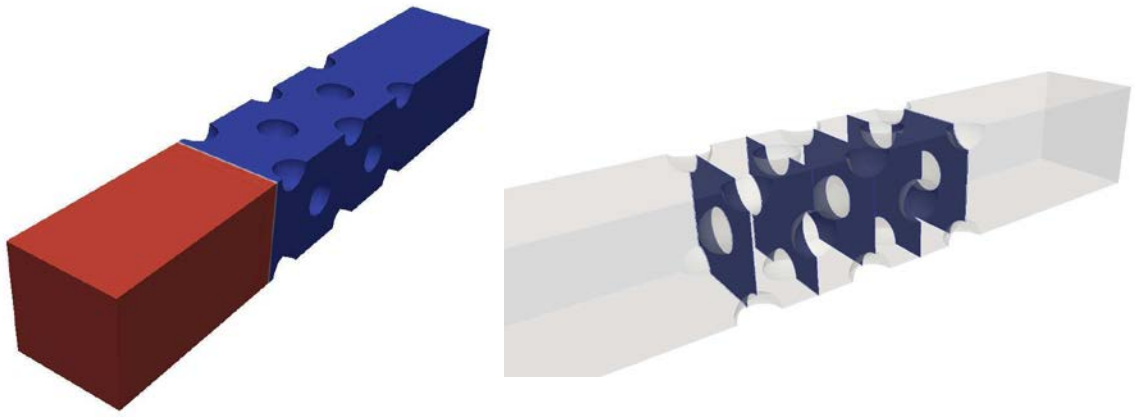


Figure 3.5. (a) 3-D simulation setup. (b) Cut sections illustrating location of particles in the simulation domain.

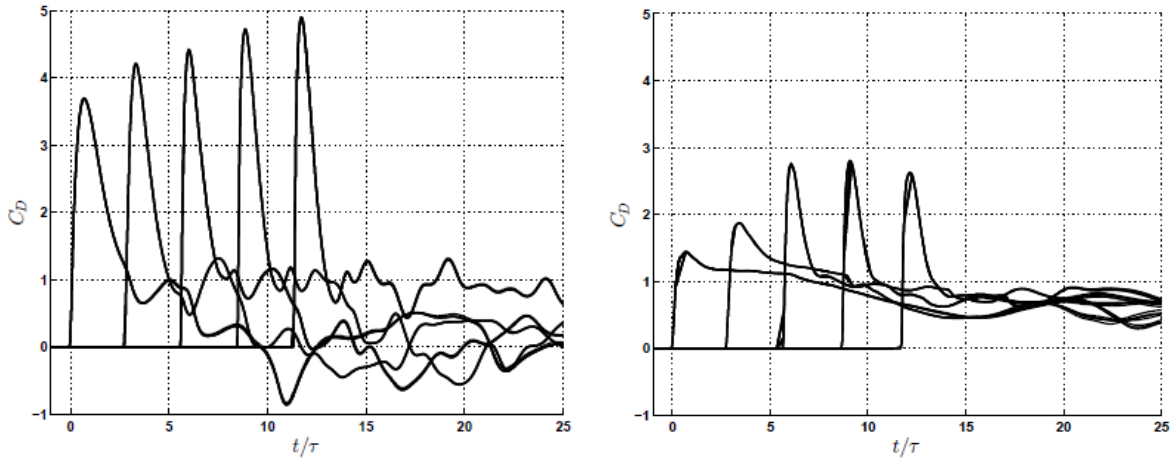


Figure 3.6. Plot of non-dimensional drag (C_D) against non-dimensional time (a) for M_s=1.5 and and ϕ=10% (b) M_s=6.0 and ϕ=10%.

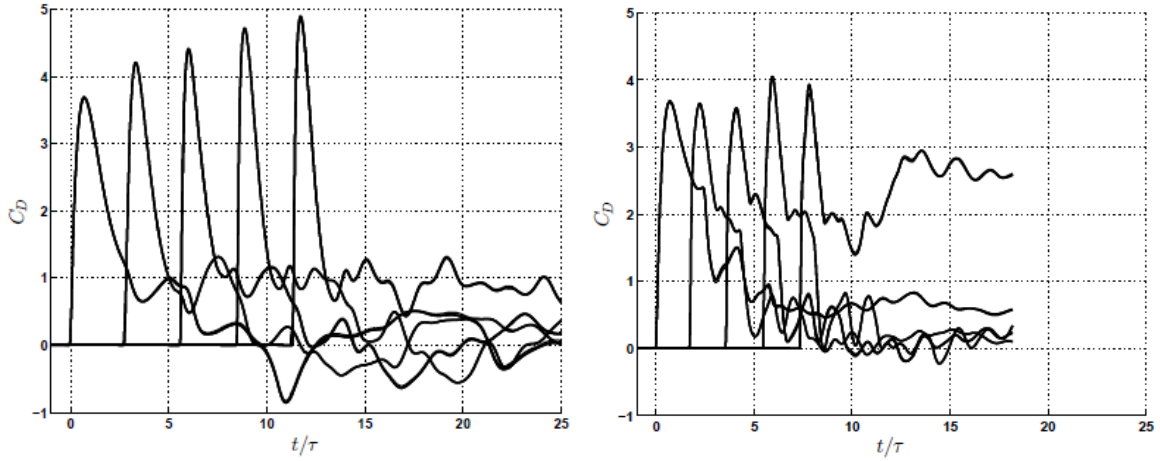


Figure 3.7. Plot of non-dimensional drag (C_D) against non-dimensional time (a) for M_s=1.5 and φ=10% (b) M_s=1.5 and φ=40%.

The case labelled RUN1 (M_s=1.5 and φ=10%) shows the same trend as the one-dimensional array, where there is increase in the successive peak drag force (see Figure 3.6a), while for the case RUN4 (M_s=6.0 and φ=10%), peak drag starts decreasing after the third plane of particles. For RUN4, the post shock flow is above critical, which results in a bow shock formation in front of the particles. The bow shocks dissipate energy from the main shock and hence weaken it which results in lower peak drag values for subsequent particles. The case RUN13 (M_s=1.5 and φ=40%) with high volume fraction does not show the same trend as that of the one dimensional array. For high volume fraction cases, the particles are closer to each other resulting in a remarkably different wave interaction and hence different trends for that case.

Shock Interaction with Random Particle Beds

The focus of the recent work on microscale shock particle interaction involving random particle beds has been to extract data for modeling particle forces. The simulations that were run on the DOE machines focused on examining four random particle beds that had global volume fractions of 10%, 15%, 20%, and 25%. These volume fractions were selected to have volume fractions that were higher than the standard dilute volume fraction limits where theory has a strong foundation. The volume fraction range was also selected to be characteristic of the type of packing fraction that may be seen in realistic particle packing conditions. The continuing work on the random particle beds has been to improve our ability to characterize the particle beds and the extract additional relevant data from the large data sets that were generated by the simulations.

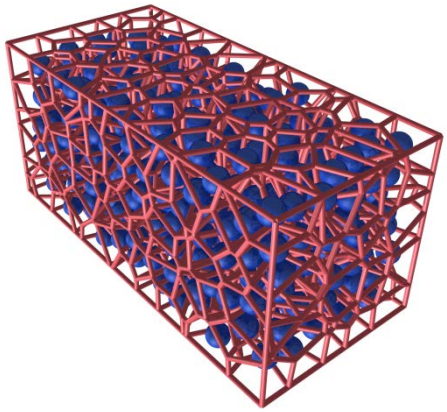


Figure 3.8. Voronoi tessellation of a random particle bed with 500 particles and a global volume fraction of 25%.

Particle Bed Characterization

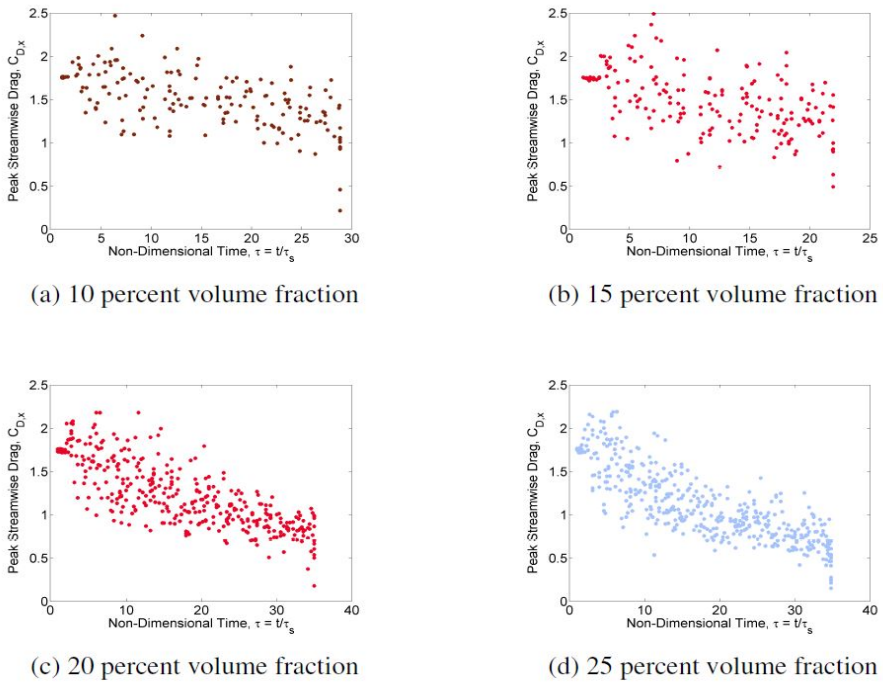
The random particle beds were generated using a rule that guaranteed that the global volume fraction of particles within the beds were 10%, 15%, 20%, and 25%. The particle distributions were random, and therefore there are regions within the particle beds where the local volume fraction deviates noticeably from the global volume fraction. We developed a method to generate Voronoi tessellations (see Figure 3.8) of the random particle beds to extend our characterization of the random particle beds beyond using only the global volume fraction. These tessellations generate small volumes, or cells, around every particle, and the volumes provide information about how tightly packed a particular particle in the bed is with respect to its neighbors. With the Voronoi cells we can estimate the local variation of the particle volume fraction within the random particle beds to provide a more complete picture for the characterization of a random particle bed.

Particle Forces

The simulations of the shock interaction with the random particle bed provides data about the total inviscid force that every particle in the bed feels during the passage of the shock over the bed. The quantity used to compare all of the particle forces was the non-dimensional drag coefficient that is normalized by the post-shock properties (see Figure 3.9).

The drag coefficient histories shown in Figure 3.10 are for particle beds with on the order of hundreds of particles. We observed a marked decline in the drag coefficient that particles deeper into the bed (further to the right on the plots) felt compared to the particles at the front of the bed.

The peak drag coefficients of the particles were also examined to reduce the complexity of the data that is shown in the total drag coefficients time histories (see Figure 3.10).



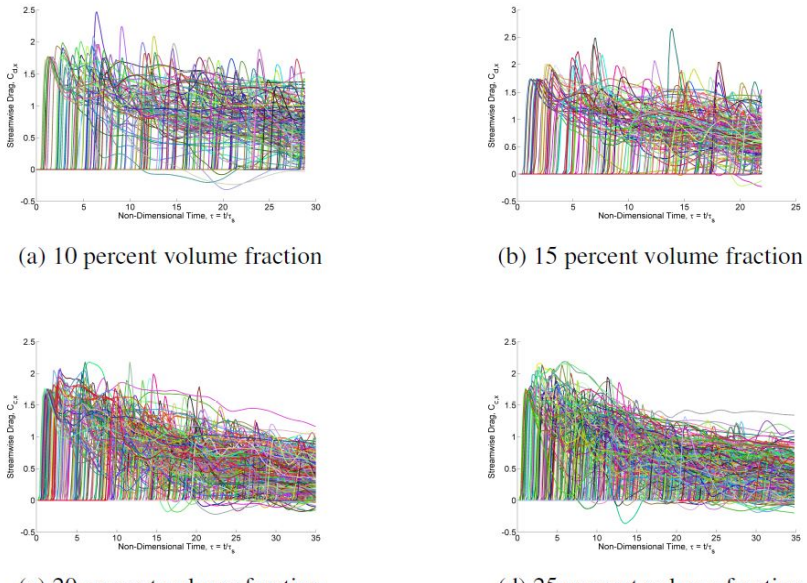
(a) 10 percent volume fraction

(b) 15 percent volume fraction

(c) 20 percent volume fraction

(d) 25 percent volume fraction

Figure 3.9: Plot of streamwise peak drag coefficients for all particles in the random particle beds for all global volume fractions.



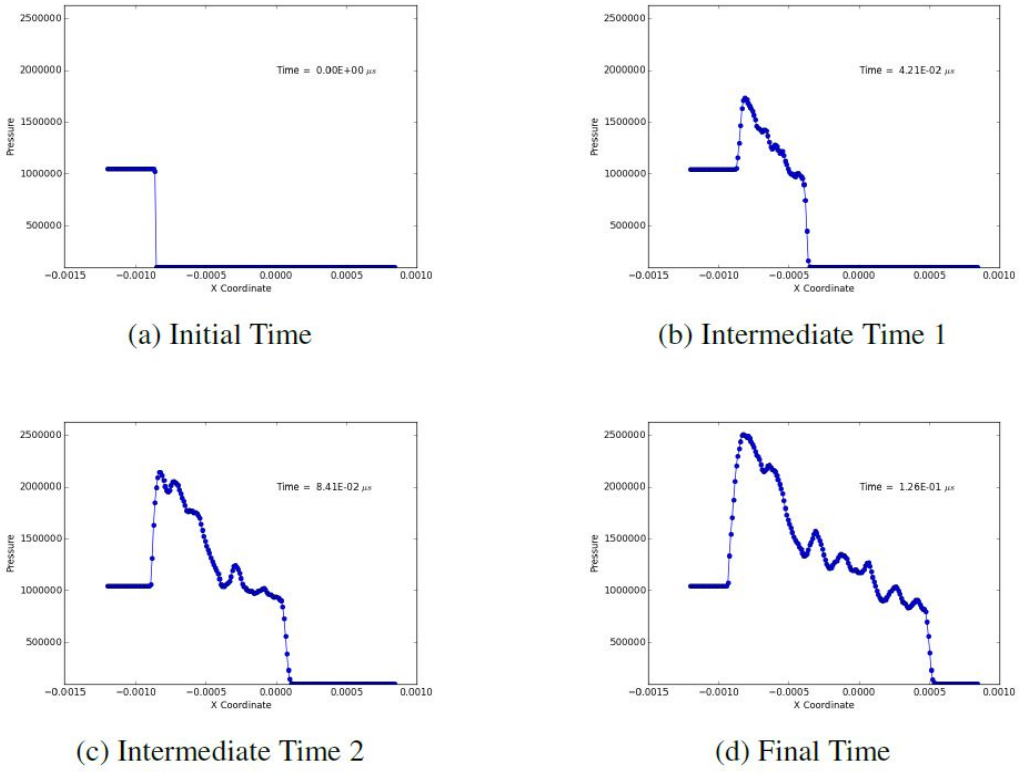
(a) 10 percent volume fraction

(b) 15 percent volume fraction

(c) 20 percent volume fraction

(d) 25 percent volume fraction

Figure 3.10: Plot of streamwise drag coefficient histories for all particles in the random particle beds for all global volume fractions.



(a) Initial Time (b) Intermediate Time 1

(c) Intermediate Time 2 (d) Final Time

Figure 3.11: Plot of streamwise average pressure variation within the random particle bed for a shock Mach number of 3.0 and a global particle bed volume fraction of 10% .

The particle beds with higher volume fractions show signs of stronger particle shielding i.e. the particles deeper into the bed do not see the same flow as the particles at the leading edge of the bed. This investigation has only looked at the sensitivity of the shielding with respect to the volume fraction at a shock Mach number of 3.0. Additional work is being done to extend the design space of the simulations to include additional shock Mach numbers.

Flow-field Interaction with Particles

In addition to investigating how the interaction with a Mach 3.0 shockwave with a particle bed affects the forces felt by the particles, we also looked at characterizing how the flow reacts to the presence of the random particle bed. Without particles the shockwave will travel uninterrupted at a constant speed in a planar geometry case.

Figure 3.11 shows the average pressure distribution through the particle bed at four characteristic times during the shock passage through a 10% volume fraction bed. We observed the formation of a large coalesced bow shock in front of the bed. This shows up on the pressure plots as a very large initial peak moving from left to right. It is difficult to see directly from the plots above, but the pressure at the shock front (the first pressure jump, approaching from the right) drops below

the original post-shock pressure. This is suspected to be the reason why the peak drag coefficients on the individual particles decay with depth into the particle bed. In addition to examining the average flow field variation as a function of depth into the particle bed, we also took a closer look at the shock front itself. The shock front was tracked by estimating its location based on the first pressure jump detected from the pre-shock conditions. The flow field variables at the shock front were observed. A parameter of interest was the pressure at the shock front as a function of time. From this data we extracted a total pressure drop between the original post-shock pressure and the measured post-shock pressure at the front. Table 3.2 shows the ratio of the pressure at the front to the original post-shock pressure.

Table 2. Ratio of the pressure at the front to the original post-shock pressure.

Volume Fraction %	Pressure Ratio
10	0.759
15	0.671
20	0.664
25	0.579

3.3 Third Year Plans

Simulation matrices not completed in year 2 will be completed in year 3.This includes all work for structured array of particles, as well as our initial effort for random packs of spheres. Numerous archival quality papers are currently being written, and year 3 will see all year 2 work submitted for publication in archival journals. We will also continue our effort at developing point-particle force models that can be used at the macroscale.

4. Experiments

4.1 ASU Experiments

4.1.1 Goals and Motivation

Multiphase studies involving compressible flow are very complicated. The gas phase may move separately from solid particles. The particles may affect the gas flow and the particle structures may also be highly influenced by the gas phase. In addition, it's possible for the particles to generate turbulence.

The computational model required to describe this type of flow must be highly complex. There is a great need, especially in the early stages of the model development, for a simple experiment that can be used for early validation of the computational codes developed by the PSAAP center. This experiment will involve simpler physics than the spherical charge capstone experiment.

The experimental team at Arizona State University proposed a simple shock tube experiment with a particle bed at the base. The shock tube will be initially separated into two regions: a lower region at atmospheric pressure (~101kPa) where the particle bed is contained, and an upper region close to vacuum (~1kPa). When the two sections merge, a shock wave is formed and the particle bed will expand upward into the lower pressure region.

The idea of designing a simple experiment like this one, is that we can examine the real physics of the flow. Dr. Kirsten Chojnicki's Ph.D. research at Arizona State University involved looking at shock velocity, mixture velocity, and the particle drag coefficient, in a similar shock tube experiment to the one proposed by ASU. In addition to looking at these quantities, we'd like to examine the gas and particle velocity fields, and the particle-gas interfacial instabilities.

By carefully designing the experiment and doing many tests, we believe we can reduce the scatter seen in Dr. Chojnicki's data. We can do this by more precisely controlling particle properties, including the size distribution. Usually, when labeling the diameter of the diameter of the particles, the expected variation is as high as plus or minus the mean. We can decrease the size distribution by further sieving the particles.

We are also interested in incorporating Schlieren imaging into our experiment. Schlieren experiments use a point source of light to illuminate fluid flow. As the pressure changes throughout the flow, the density changes—meaning the index of refraction also changes. In Schlieren experiments these inhomogeneities in density can be explored by examining the light intensity differences appearing on the camera images.

4.1.2 Description of Experiment

The basic profile of the shock tube will be a six foot tall glass tube with a six inch diameter circular footprint. A particle bed will be placed at the bottom of the shock tube. The particle bed will be composed of small glass spheres. The glass spheres will have diameters between 90 and 150

microns. A mylar diaphragm will be used to separate the high pressure (~101kPa) from the low pressure region (~1kPa).

The experiment was designed in this way to maximize optical access to the flow. The wide windows will allow for particle image velocimetry measurements and Schlieren measurements. Using optical techniques, we can measure the contact line velocity, the gas velocity, and the particle volume concentration. We can also investigate the particle bed interface and how it evolves in time. The main parameters that will be used to vary the experiment will be the initial pressure ratio (typically referred to as p4/p1) and the particle diameter.

The diaphragm separating the low pressure region from the high pressure region will be put well above the particle bed. This will ensure the diaphragm does not interfere with the expansion of the particle bed. The sidewalls of the shock tube will have many PCB Piezotronics Model 113B22 pressure sensors. They have a sensitivity of +/- 10% and output 0.145 mV/kPa. Using these pressure sensors, we will be able to track the shock wave as it travels upward into the test section. We will also be able to examine the pressure fluctuations along the length of the shock tube, including the particle bed, during the experiment.

4.1.3 Accomplishments during FY 2014

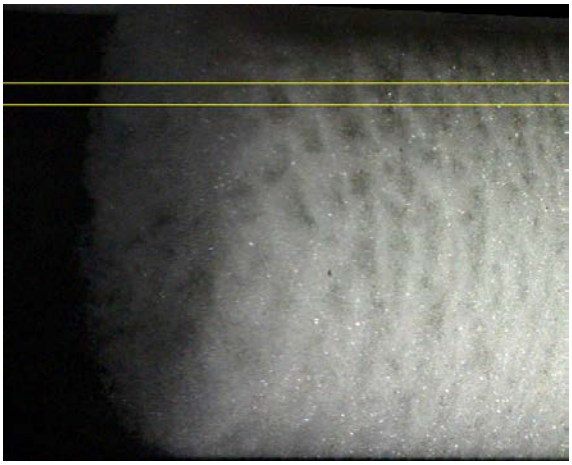
Fourier analyses of 1D data and of images can extract patterns and frequencies (see Figure 4.1). 1-D and 2-D Fourier transforms were used to analyze frequencies in the cell-like structure which develop in the particle bed expansion. An intensity lineout, which averages the intensity along the Y-direction, was analyzed using a 1-D Fourier transform. A series of frequency peaks were extracted. The higher the frequency peak (bottom left image), the more intensity is associated with that spatial frequency. Because the cells grow in time and then continue to form deeper into the particle bed, this frequency measure can serve to measure the rate of cell growth (an Eulerian measure of a Lagrangian quantity).

A similar analyses can be performed by examining an image, as opposed to an intensity lineout, and using a 2-D Fourier transform (see second column of images). Pulling detailed data out of this analyses is more difficult, as each individual particle causes changes in intensity. For this reason, the image was convolved with a Gaussian filter. After the image was filtered, the 2-D Fourier transform was applied. The results are shown as a colormap and as a surface plot. Similar features seen in the intensity lineout on the left column are seen when looking at the profile of the surface plot.

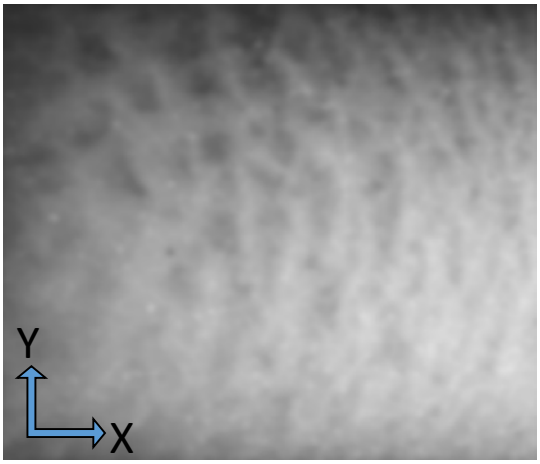
By using tools like this, we can quantify the cell distribution frequency, which will help determine an origin for these features. Fourier analyses of intensity lineouts and of images will be further explored as we get more data.

Figure 4.1: Fourier Analyses

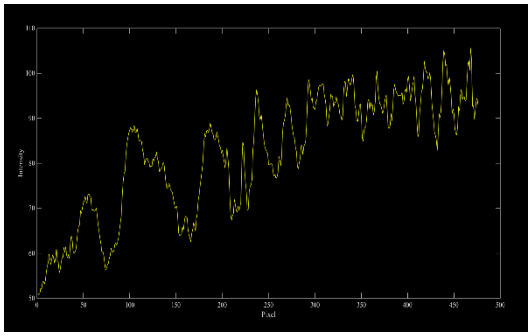
Location of Intensity Lineout



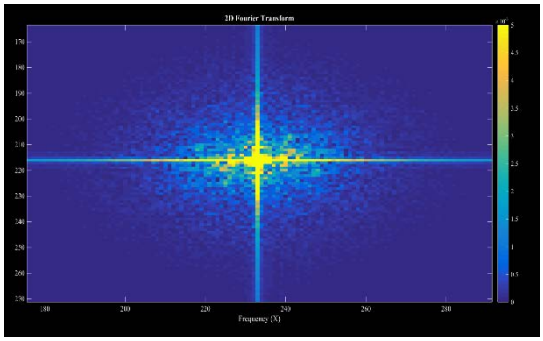
Gaussian Blurred Image



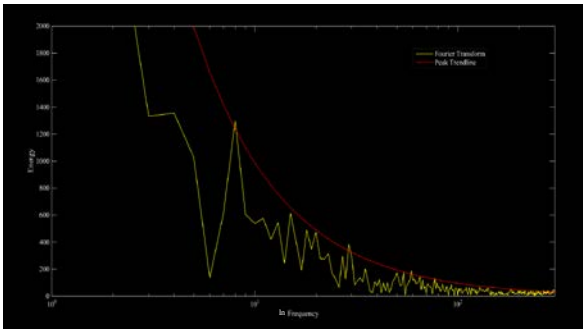
Intensity Lineout



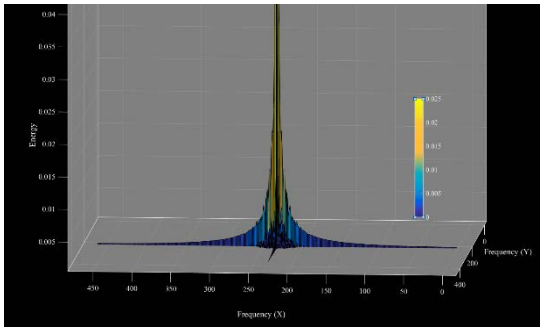
2D Fourier Transform Colormap



1D Fourier Transform with Trend line



2D Fourier Transform Surface Plot



$y = 10159x^{-1.012}$

Goodness of Fit (R^2) = 0.9878

Changes to the shock tube design has been considered after extensively examining the equations for reflected shock waves and expansion fans. In the case where the height of the low pressure region (p1) is three meters, the reflected shock wave will reach the diaphragm in 0.0117 seconds, as shown in Table 4.1. This is important because the reflected shock wave could potentially alter the air flow and the particle bed movement. The team at ASU has considered increasing the height of the shock tube design for this reason. The taller the low pressure section, the longer it will take for the reflected shock wave to return and possibly interfere with the flow. The goal of this experiment is to provide the simplest case for a shock tube and expansion fan, therefore making the shock tube taller may be desirable.

Reflected Shock Wave (air only)		
M_r/(M_r^2 - 1)	0.579020077	
M_r	y, goal seek to 0	
2.184128312	-0.00098147	
U_reflected	320.9035738	[m/s]
U_shock	1269.14778	[m/s]
Length of p1 region, L1	3	[m]
Time before reflection	0.011712394	[s]
t = 0 is diaphragm rupture		

Table 4.1: Timings for reflected shock wave

It may also be desirable to change the height of the particle bed. The expansion fan also has the potential to reflect off of the bottom of the shock tube. If the high pressure region (p4) is 0.1 meters, then it would take only 0.00029 seconds (as shown in Table 4.2) for the expansion fan to reflect off the bottom of the shock tube (in the absence of particles). Minimizing complicating reflected shock waves and expansion fans will help the experimental team align our experiments with the theoretical team's simulations.

Expansion Fan (air only)		
Length of p4 region, L_4	0.1	[m]
gamma	1.4	
p1	1	[kpa]
p2	15.80620265	[kPa]
p3		[kPa]
p4	101	[kPa]
p4/p1	101	
p1/p4	0.00990099	
M_shock, table value	3.7001393	
speed of sound, a1	343	[m/s]
speed of sound, a2		[m/s]
speed of sound, a3	146.9252388	[m/s]
speed of sound, a4	343	[m/s]
Contact line velocity, u_p	-980.373806	[m/s]

Table 4.2: Timing for reflected expansion fan

The experimental team at ASU had previously been using an older camera, which worked sufficiently well for examining general large scale flows, as long as the bulk motion of the material was the object being measured. The maximum frame rate with reasonable resolution was 1,000 frames per second (fps).

A new camera was purchased which is a Phantom v641 from Vision Research. It is shown in Figure 4.2 along with a table showing some of its features.

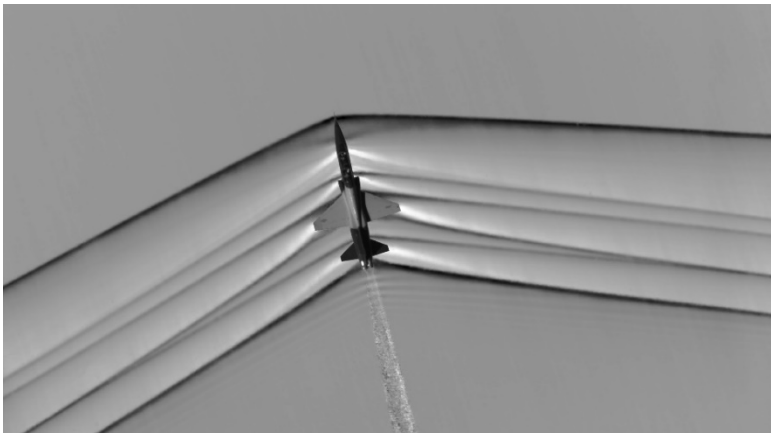
Figure 4.2: Phantom v641

Vision Research (Phantom) v641	
Resolution	2560 x 1600
Frame rate (at max resolution)	1450 fps (best for pixel count)
HD resolution (1080p)	1920 x 1080 up to 2,560 fps
HD resolution (720p)	1280 x 720 up to 5,350 fps
Max frame rate	219,000 fps (256 x 8)
Bits	12 bits
Internal memory	16 GB
High G (impact survivability)	Up to 30G
FAST data offload to laptop	10 GB Ethernet (7.68 GB per minute)
Warranty	2 years
Resolution at 10,000 fps	600 x 800
ISO Sat	16,000T; 6,400D

Several experiments were performed using the new camera in attempts to better understand the expansion wave as it traveled downward into the particle material. It was decided that we would try to image the expansion fan using a technique called Schlieren imaging. Schlieren imaging takes advantage of the fact that the way light travels through air is dependent on the density of the air. If the air is denser, light is refracted more than if the density is lower. If there is an interface between two fluid parcels of different density, it appears on the camera as a bright or dark region (the light is either refracted away from the pixels on the camera, yielding a dark region; or closer to the pixels, yielding a bright region). See Figure 2 for an example of Schlieren imaging.

The ASU team was able to capture some faint features in a high-speed video. These faint changes in intensity represent the density changes in the air. The changes in intensity will be analyzed and quantified, so that a velocity of the expansion wave can be determined. These values will be compared to theoretical values.

Figure 4.3: Schlieren imaging of a jet. The interfaces between different density air parcels appear as dark lines.



4.1.4 Third Year Plans

During the next year, the experimental team at ASU will begin building the large shock tube. Many parts have already been purchased and sent to the machine shop to be processed. The old shock tube was disassembled and the parts that are compatible with the new design will be re-used to save money.

ASU will further investigate the flow processes using Schlieren Imaging to get better sensitivity and image quality for the shock wave and expansion fans seen in these experiments.

4.2 Eglin AFB Experiments

The data and information in this section of the report is Distribution A. Public release, distribution unlimited. 96ABW-2015-0133. 96ABW-2016-003.

4.2.1 Goals and Motivation

The primary goal of the experiments conducted at Eglin Air Force Base is to provide validation quality data at the micro, meso, and macroscales. AFRL/RW has completed initial experiments at the micro- and meso-scales as described in this section, and the data have been given to the UF-CCMT modeling team for uncertainty analysis. In years 3 and 4 of this effort, AFRL/RW will perform:

a. Microscale experiments with a no-particle, detonation-only, planar geometry in support of T3, uncertainties due to thermodynamic (EOS) and transport properties;
b. Mesoscale experiments with a planar particle bed in support of T5, compaction modeling uncertainty;
c. Microscale experiments with a planar particle cluster in support of T6/T7, uncertainty in finite Re, Ma and volume fraction dependent drag and heat transfer; and
d. Macroscale experiments with an annular particle bed in support of T0, validation of the demonstration problem.

Test #	Date	Driver	Particle(s)
1	2/25/2015	400 psi	11 – 2mm tungsten spheres (0.078 g/ea)
2	2/25/2015	400 psi	Salt
3	2/25/2015	500 psi	Salt
4	2/25/2015	500 psi	Salt
5	2/25/2015	500 psi	Salt
6	2/25/2015	1000 psi	Salt
7	2/26/2015	RP83 + 3 N5	Single – 2mm tungsten sphere (0.078 g/ea)
8	2/26/2015	RP83 + 3 N5	Single – 2mm tungsten sphere (0.078 g/ea)
9	2/26/2015	RP83 + 3 N5	3 – 3mm tungsten hexes (0.402 g/ea)
10	2/26/2015	RP83 + 3 N5	Salt
11	2/26/2015	RP83 + 3 N5	Salt
12	2/26/2015	RP83 + 3 N5	4 – 2mm tungsten spheres (diamond pattern) (0.078 g/ea)

Figure 4.4. Test sequence for microscale experiments.

4.2.2 Microscale Experiments

Twelve small-scale explosive experiments were performed at Eglin AFB in February 2015. These experiments are considered microscale in that a small number of particles are of interest. The test data include shock arrival times, high-speed video, x-ray images, and witness panel results. The twelve experiments are equally divided into gas-valve tests (compressed helium, tests 1-6) and explosively-driven test (tests 7-12), as shown in Figure 4.4. Table salt was substituted for tungsten particles as noted in the table in an attempt to visualize the gas flow patterns with the fine particles acting as tracers.

Experimental Setup

The microscale experiments were conducted at the Advanced Warhead Experimental Facility (AWEF) at Eglin AFB, FL. All tests utilize the same basic experimental setup as shown in the schematics of Figure 4.5. The actual setup is shown in more detail in Figure 4.6, where the pressure probe array (left image) and x-ray heads (right image) are of interest. The pressure array is a 3x3 grid of pressure probes, slightly offset from the test centerline with the middle probe removed due to the high likelihood of fragment impact. Four x-ray heads were used to increase the ability to track the particles' location and velocity within the fireball.

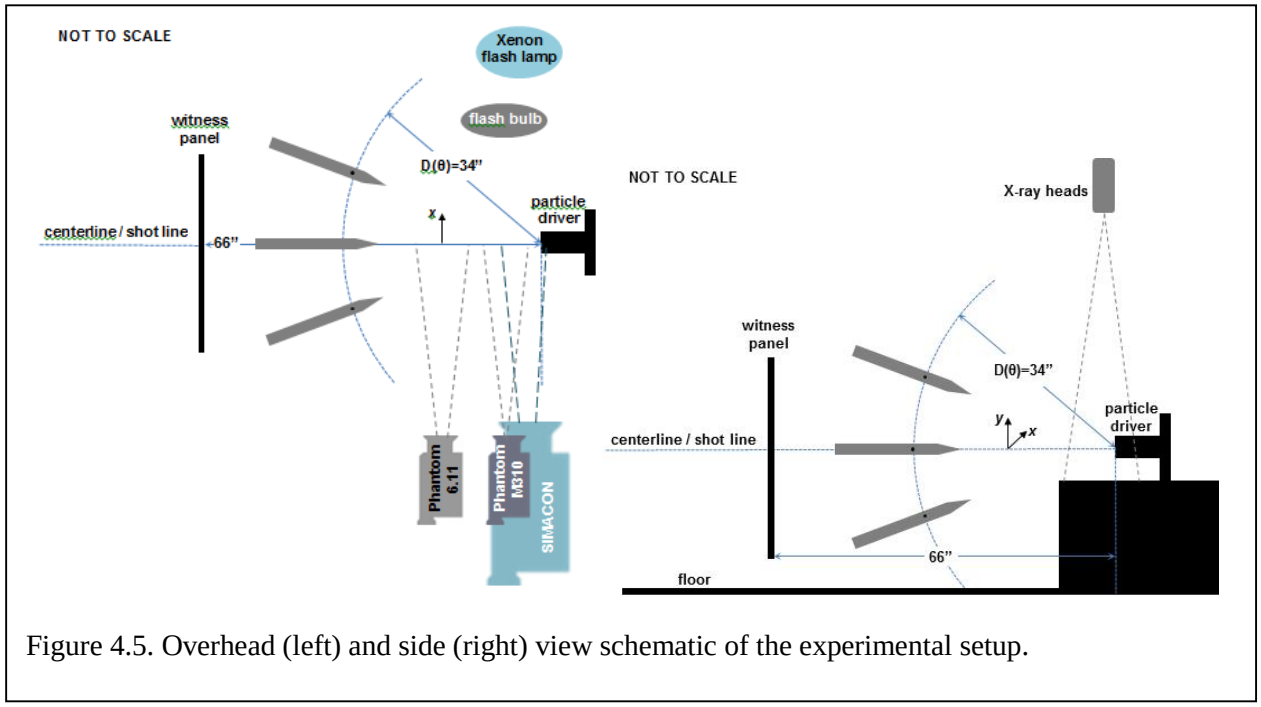


Figure 4.5. Overhead (left) and side (right) view schematic of the experimental setup.

Figure 4.6. Side view of the experimental setup with emphasis on the pressure probe array (left) and the x-ray heads (right).

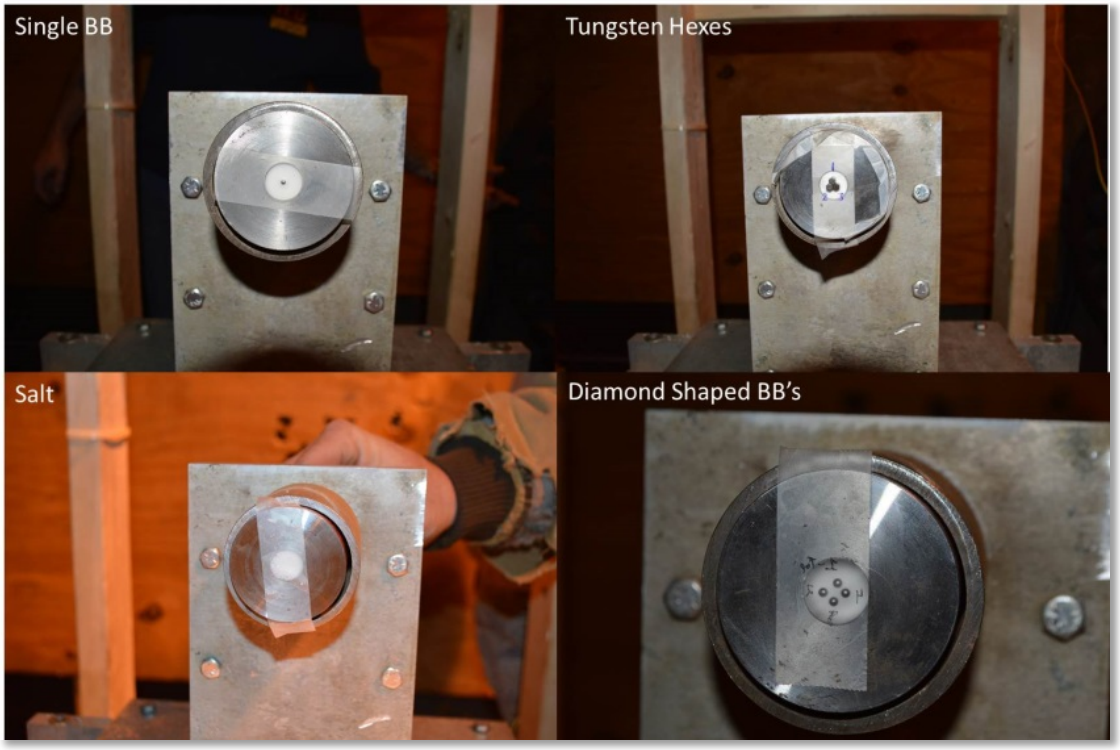


Figure 4.7. Representative microscale test configurations.

Compressed helium provides an optically transparent shock wave and was used in the gas-valve experiments refine the settings on the diagnostic equipment, particularly the cameras. During the second day of testing, the compressed helium driver was replaced by an explosive charge. For

these experiments the explosive charge consisted of three stacked N5 pellets (each 0.5” in length and diameter) initiated by an RP-83 detonator. The particle configurations noted in Figure 4.4 are shown in Figure 4.7.

Preliminary Results

Data from the experimental diagnostics are detailed below. For test 8, one of the delay generators malfunctioned, resulting in a loss of Simacon camera data. The delay generator was replaced in shot 9, but an improper arming sequence resulted in a loss of all data.

Pressure Probes

The pressure probes from the compressed helium tests show a relatively weak shock (≤1 psi) and a sharp pressure rise. Alternately, the pressure probes from the explosive tests show a much stronger shock (8-23 psi) with a complex pressure rise, exhibited in a non-noise signal, followed by an oscillation, then a sharp rise in signal.

High Speed Video

High speed images were recorded for the Phantom 6.11, Phantom Miro M310 and SIMACON. The Phantom 6.11 was used to capture the detonation products down range. A representative image from each shot for the Phantom 6.11 is shown in Figure 4.8. The SIMACON camera was used to capture the blast wave at close range, but has a faster framing rate (but limited number of images) than the Phantom Miro M310. Representative images from the SIMACON are shown in Figure 4.0. The Phantom Miro M310 was used to capture the blast wave at close range. Representative images are shown in Figure 4.10.

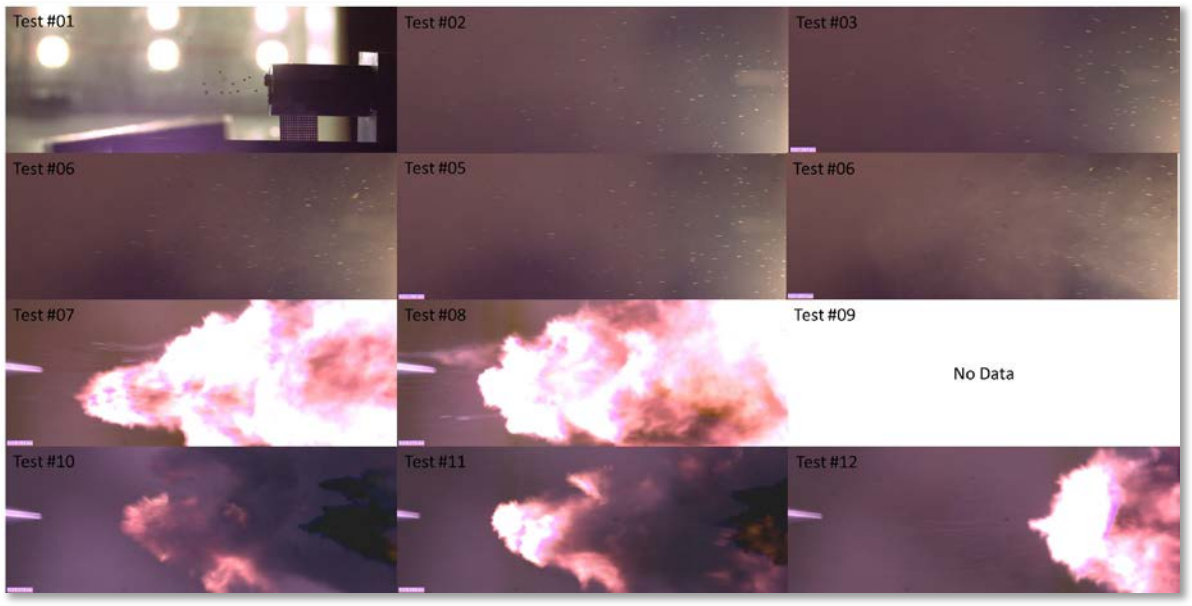


Figure 4.8. Representative Phantom 6.11 test images.

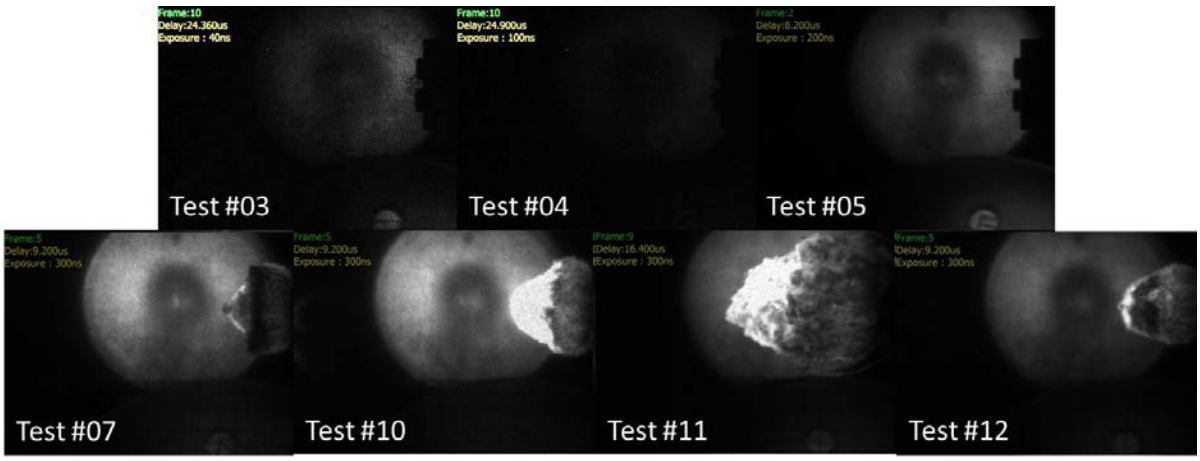


Figure 4.9. Representative SIMACON images.

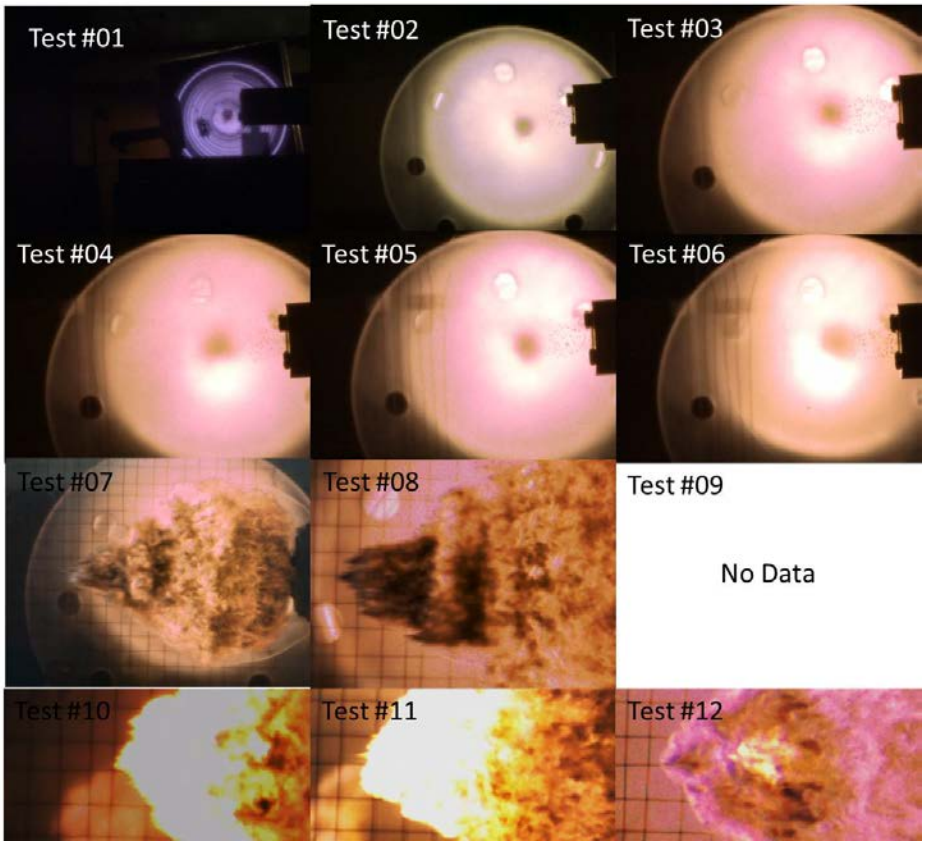


Figure 4.10. Representative Phantom Miro M310 test images.

X-ray Imaging

Four x-ray heads were mounted above the particle driver, as shown in Figure 4.5-4.6. The x-ray heads were triggered on a timing delay of 10-20 microseconds; the particle velocity is of primary interest; the velocity is shown in Figure 4.11 and the x-ray images are shown in Figure 4.12. The large number and small size of the salt particles in test 10 and 11 precluded accurate velocity measurements.

	Test #07	Test #08	Test #09	Test #12
Head 1	550	550	759	---
Head 2	651	560	751	---
Head 3	723	644	707	631
Head 4	776	576	---	806

Figure 4.11. Particle velocity (in m/s) measured from the x-ray images.

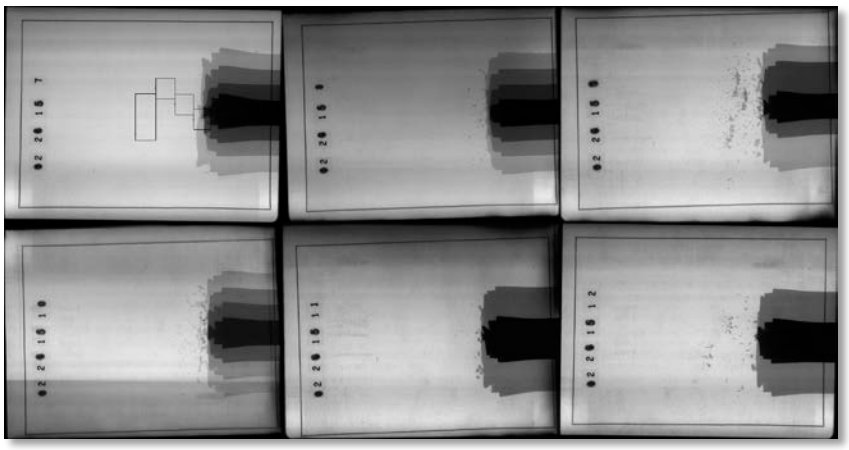


Figure 4.12. Multiple exposure x-ray images for tests 7 through 12 (left to right, top to bottom).

Witness Panels

Thin aluminum witness panels were used to determine the particles' far-field impact locations in tests 7, 8, 9, and 12. The witness panels were placed 66" from the particle driver, as shown in Figure 4.5 and the center of each panel was determined using a bore sighting laser. The final witness panels for tests 7, 8, and 12 are shown in Figure 4.13; test 9 did not show any impacts on the witness panel due to particle break-up during detonation. No particles were located or recovered. The location of each impact on the witness panels is shown in Figure 4.14.

Figure 4.13. Particle velocity (in m/s) measured from the x-ray images.

	Test #07	Test #08	Test #12
Particle 1	(-19, -46)	(7, 21)	(84, 182)
Particle 2			(-192, -148)

Figure 4.14. Location of the witness panel impacts in mm.

4.2.3 Mesoscale Experiments

Twenty-two mesoscale explosive experiments were performed at Eglin AFB in October-November 2015. The diagnostics and setup are the same as the microscale experiments. The 22 experiments are divided into gas-valve tests (tests 1-12) and explosively-driven tests (tests 13-22), as shown in Figure 4.15. The first tests were completed with a coarse steel powder. Tungsten powder was used for the remaining tests, where the best configuration of Tungsten required some experimentation, as seen in tests 8, 9, and 10. It was determined that the Tungsten powder was ejected most consistently when pressed in alcohol between sheets of tissue; tests 10-22 used this method.

Experimental Setup

The experimental test setup is the same as that detailed for the microscale experimental setup, detailed above.

Preliminary Results

Final data analysis is not yet complete, but has been shared with UF CCMT personnel.

4.2.4 Summary

The micro and mesoscale experiments performed at Eglin AFB, FL, present the UF CCMT staff with a first-look at diagnostic and measurement techniques. The ultimate objective is to provide the UF-CCMT with high quality data for validation and verification of their turbulent multiphase models.

Test #	Date	Driver	Particle(s)	Particle Weight (g)
1	10/28/15	1700psi	Coarse steel powder	
2	10/29/15	1700psi	Coarse steel powder	
3	10/29/15	1700psi	Coarse steel powder	
4	10/29/15	1900psi	Coarse steel powder	
5	10/29/15	1700psi	Coarse steel powder	
6	10/29/15	2000psi	Coarse steel powder	
7	11/2/15	1660psi	Coarse steel powder	0.575
8	11/2/15	2000psi	112µm Tungsten	0.035
9	11/2/15	2000psi	112µm Tungsten (capsule)	0.88
10	11/2/15	2000psi	112µm Tungsten (pressed tissue)	0.505
11	11/2/15	2000psi	112µm Tungsten (pressed tissue)	0.5
12	11/2/15	2000psi	112µm Tungsten (pressed tissue)	0.245
13	11/3/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
14	11/3/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
15	11/3/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
16	11/4/15	RP-2, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
17	11/4/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
18	11/4/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
19	11/5/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
20	11/5/15	RP-80, 1 pellet N5	112µm Tungsten (pressed tissue)	0.5
21	11/5/15	RP-80, 2 pellets N5	112µm Tungsten (pressed tissue)	0.5
22	11/5/15	RP-80, 2 pellets N5	112µm Tungsten (pressed tissue)	1

Figure 4.15. Test sequence for mesoscale experiments.

5. UB Team

5.1 Overview

The primary objective of the UB team is to reduce the uncertainty in both simulations and experiments to a level that would allow meaningful validation of the simulation code. A secondary objective is to help other teams quantify and reduce the uncertainty in their applications. Extensive efforts are required for uncertainty detection and uncertainty quantification effort, which poses two challenges. First, the UB team has to work collaboratively on detecting large uncertainties with the micro and meso/macro physics and experiment teams as well as the exascale team. Second, the computational requirements for good uncertainty quantification are orders of magnitude higher than simulation alone. To address the first challenge, we selected problems that were ready to tackle with the various teams. We selected the mesoscale shock tube configuration to establish working relationship and procedures to collaborate with the physics and experimentalist teams, mostly because experimental results for that scale are already available. To address the computational challenge we have been evaluating and developing several surrogate based methods as UQ tools. We also collaborate with the micro physics team to carry out convergence study of the miscroscale simulation. With the exascale team, we chose a problem of predicting matrix multiplication execution times for collaboration.

5.2 Validation, Uncertainty Quantification and Uncertainty Budget of Mesoscale Shock Tube Simulation

As a preliminary study for subscale and overall validations, we are engaging to validate the mesoscale simulation, the shock tube with a particle curtain. Figure 5.1 (a) shows the configuration of the experiment where a shock is created from high pressure gas in the driver section into the driven section by bursting a diaphragm between two sections. The shock interacts with a particle curtain in the test section as is illustrated in Figure 5.1 (b). We plan to validate the capability of predicting the motion of the particles due to the shock by comparison with experimental observations; the upstream and downstream front positions of the particle curtain. However, the

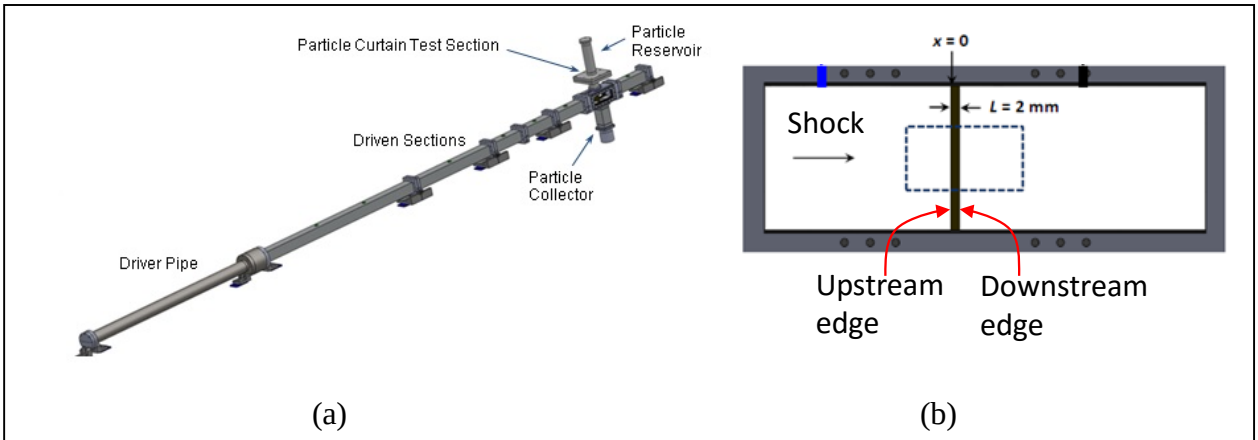


Figure 5.1. Schematic figure of the shock tube (a) and particle curtain in test section (b).

initial particle positions were not measured from the experiment while the positions have to be defined in the simulation. Thus the average front positions of the particle curtain over the initial particle position variation are used as the prediction metrics (PM).

Important outcomes of validation are estimation of the prediction error and quantification of its uncertainty. Figure. 5.2 shows errors and uncertainty sources in out validation process. The measured PM and the calculated PM are compared and the discrepancy is the prediction error due to physical model, numerical model and discretization. As a first step in uncertainty quantification (UQ), we identified and classified uncertainty sources. The uncertainty in the prediction error is due to uncertainty sources, including the measurement uncertainties in input and PMs and sampling uncertainties due to a finite number repeated experiments and simulation runs to obtain the average prediction metric. By comparing the uncertainty in the prediction error estimates to the discrepancy between simulation and experiment, we also get an idea of the possible range of magnitudes of the prediction errors.

The prediction error of the 1-D simulation code (Rocflu lite) was thus quantified. Figure 5.3(a) shows predicted and measured PMs with their uncertainties of 95% confidence in terms of time. The width of the band represents the uncertainty in the simulation side and the band says that the true edge locations will be within the bands with 95% probability. The variations between symbols from measured PMs represent the uncertainty in experiments. Figure 5.3(b) shows the discrepancy between calculated PMs with UQ. The width of the band represents the uncertainty in the validation process.

The simulation was examined using groups of runs to identify anomalies in simulation consistency. Different arrangements of the parameter space points allowed us to uncover anomalies and noise

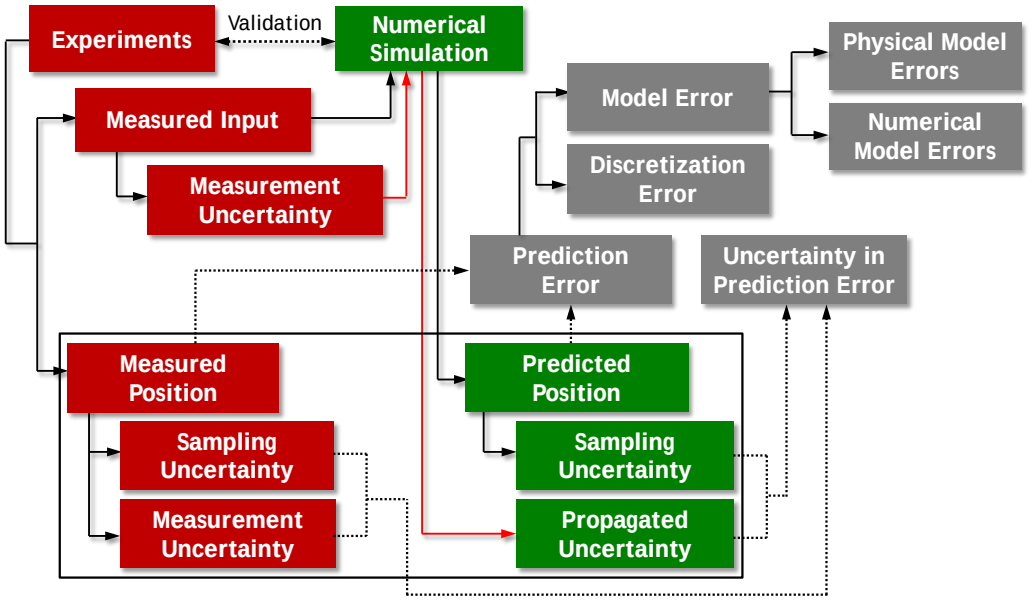
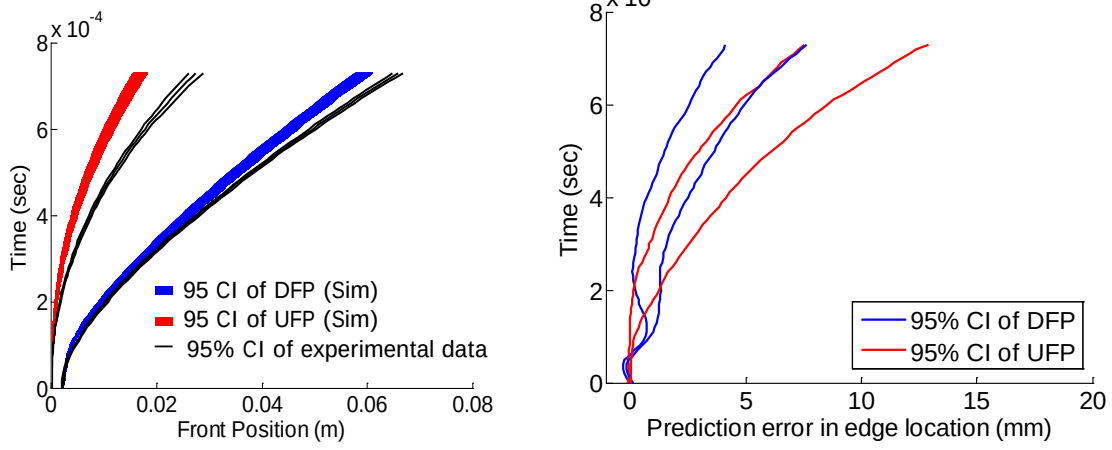


Figure 5.2. Sources of uncertainty and errors in validation process

in the simulations. Particle curtain volume fraction, curtain thickness and particle diameter were selected as the parameters since the variations of the parameters affect most the PMs. Investigating the discrepancies, the physics team, found that a gas flux calculation model (AUSM+) was a large error source of the simulation. Based on the discovery, improvement of the model was made AUSM+ upgraded to AUSM+up which was developed for multiphase flow problems. The detailed description of the anomaly detection is presented in the following section.

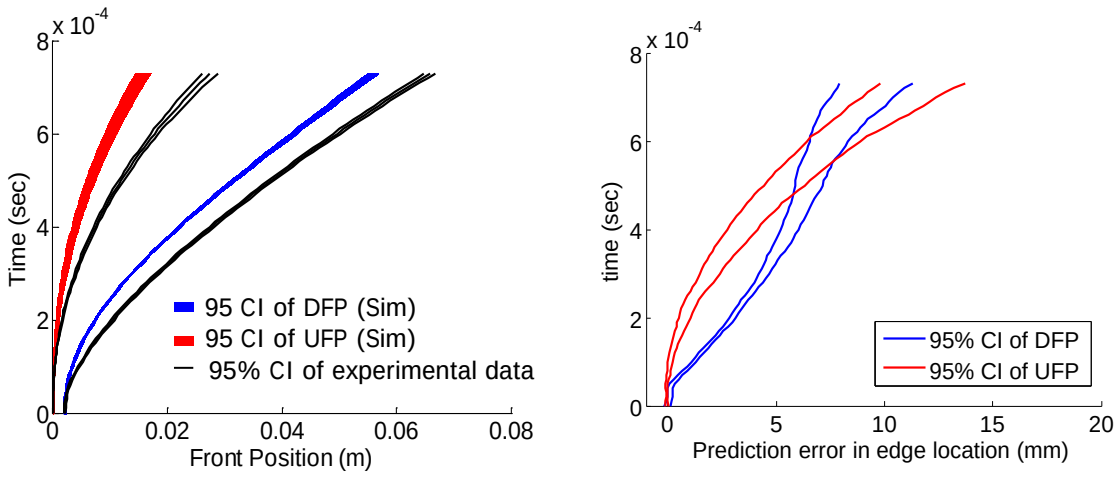
Figure 5.4 shows a comparison between the simulation with the improved model and the experiments and the corresponding prediction error estimation. An interesting observation from Figure. 5.4(a) is that the model error reduction through the model improvement increased the discrepancy compare to that in Figure 5.3(a). This tells that the error in AUSM+ may have been compensating for other errors. It gave us the wrong impression last year that the disagreement between simulation and experiment (prediction error) was small. This provides a lesson indicating that good agreement between simulations and experiments is not always a good measure of the prediction capability. The model improvement also reduced uncertainty that is reflected in the bandwidth reduction of the downstream front location.

To quantify the uncertainty source contributions of the improved simulation we have launched an uncertainty decomposition effort. Figure 5.5 shows the contributions of uncertainty sources to the uncertainty in prediction errors: sampling uncertainty, experimental sampling uncertainty and propagated uncertainty. Since, in the validation and UQ framework, the uncertainties are independent, the contributions were calculated using their variances. The portion of individual uncertainty source is the ratio between the variance of the source and the total variance. The contributions of individual uncertainties are presented in different colors and the bands of the plots



(a) (b)

Figure 5.3. A comparison between predicted and measured particle front positions (PMs). (a) Predicted and measured PMs with uncertainties of 95% confidence. (b) Prediction errors and quantified uncertainties of 95% confidence.

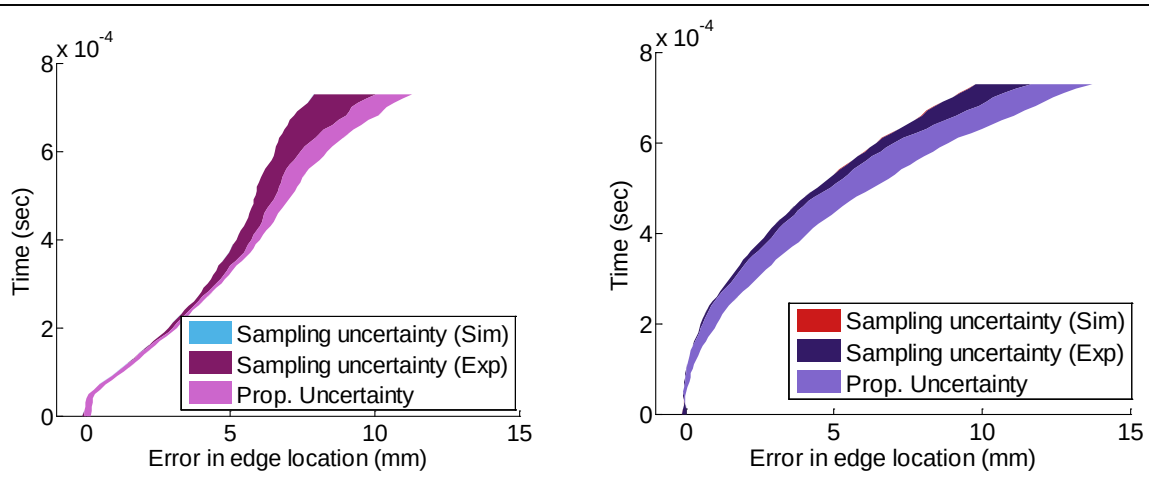


(a) (b)

Figure 5.4. A comparison between predicted and measured PMs with the modified model (AUSM+up). (a) Predicted and measured PMs with uncertainties of 95% confidence. (b) Prediction errors and quantified uncertainties of 95% confidence.

in Figures. 5.3(b) and 5.4(b) are decomposed into different colors as Figure. 5.5. We found that the propagated uncertainty affects most the uncertainty in the prediction error.

A graduate student, Giselle Fernandez, studies 1-D shock tube simulation with help of Bertrand Rollin who is in charge of macro the 1D shock tube simulation. We have also interacted with Justin Wagner in Sandia National Labs to identify and understand uncertainty sources in shock tube experiments and we identified uncertainty sources. This interaction is now extended to include another graduate student, Sam Nili, and an undergraduate Justin Mathew.



(a) (b)

Figure 5.5. Prediction errors and uncertainties with AUSM+up (a) upstream front position. (b) downstream front position.

5.3 Simulation Anomaly Detection using Groups of Runs

(Giselle Fernandez leading this effort). In order to understand and identify uncertainty sources a design of experiment (Full factorial design with 3 factors and 4 levels) was performed varying the parameters identified to have the largest effect on uncertainty in the quantities of interest, particle diameter (D), initial particle curtain thickness (t) and particle volume fraction (PVF). It was found that high PVF values (around 28%) usually led to negative pressures. With the help of the physics team I implemented two new tools in the 1D simulation code in order to have better statistics about the location of the particle curtain edges. The first one is the possibility of a random initial particles position and the second one is the possibility of make an average of the n first and last particles.

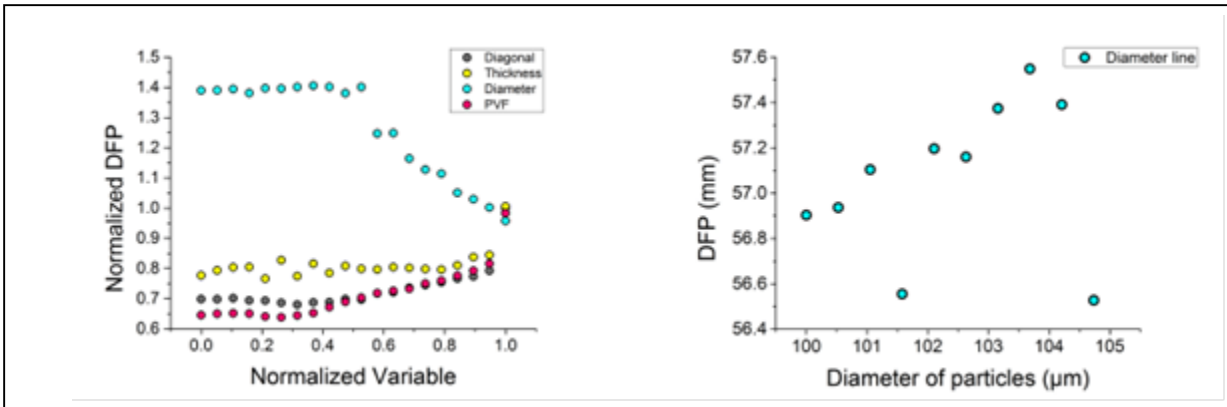


Figure 5.6. DFP vs. parameter variation for AUSM+ scheme

Using the method of converging lines (mostly used in extrapolation to improve the predictions, Zhang, et al., 2015) as a sampling technique, with an outlier point as a target point, a high level of noise was detected and contrary to what is expected, DFP moves faster as the particle diameter increases in some regions as it can be seen in Figure 5.6.

Due to the anomalies and negative pressure errors found in the code version, it was clear that updates were needed to the simulation. Fortunately, the simulation team was already aware of

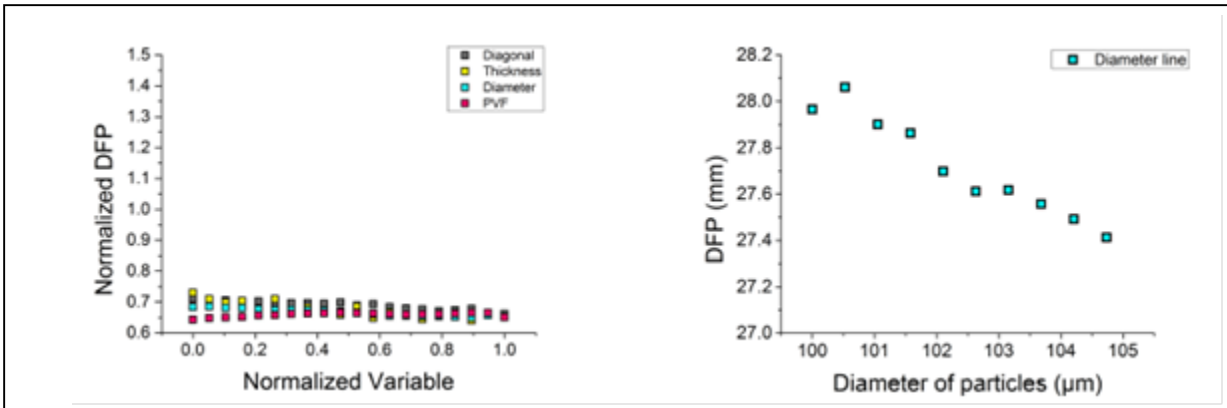


Figure 5.7. DFP vs. paramer variation for AUSM+up scheme.

some of the errors, which were attributed to a deficient flux scheme used (AUSM+ scheme), and were working towards a multiphase flux scheme implementation, AUSM+up. After the implementation of AUSM+up, no outliers were found in the uncertainty domain, the noise level was reduced and DFP moves slower as the diameter increases as shown in Figure 5.7.

5.4 Extrapolation

(Yiming Zhang leading this effort). In the CCMT center, there are many challenges requiring extrapolation. One of the goals of our team is to provide technical support for extrapolation to tackle these challenges. We collaborated with the exascale team to assess computation cost of matrix multiplication for various size matrices. With the microscale team, we worked to predict computation time of CFD simulation in terms of number of elements in the discretized domain.

We carried out studies for extrapolation using the matrix multiplication function problem, where we need to predict computation times for large size matrices from computation times for smaller

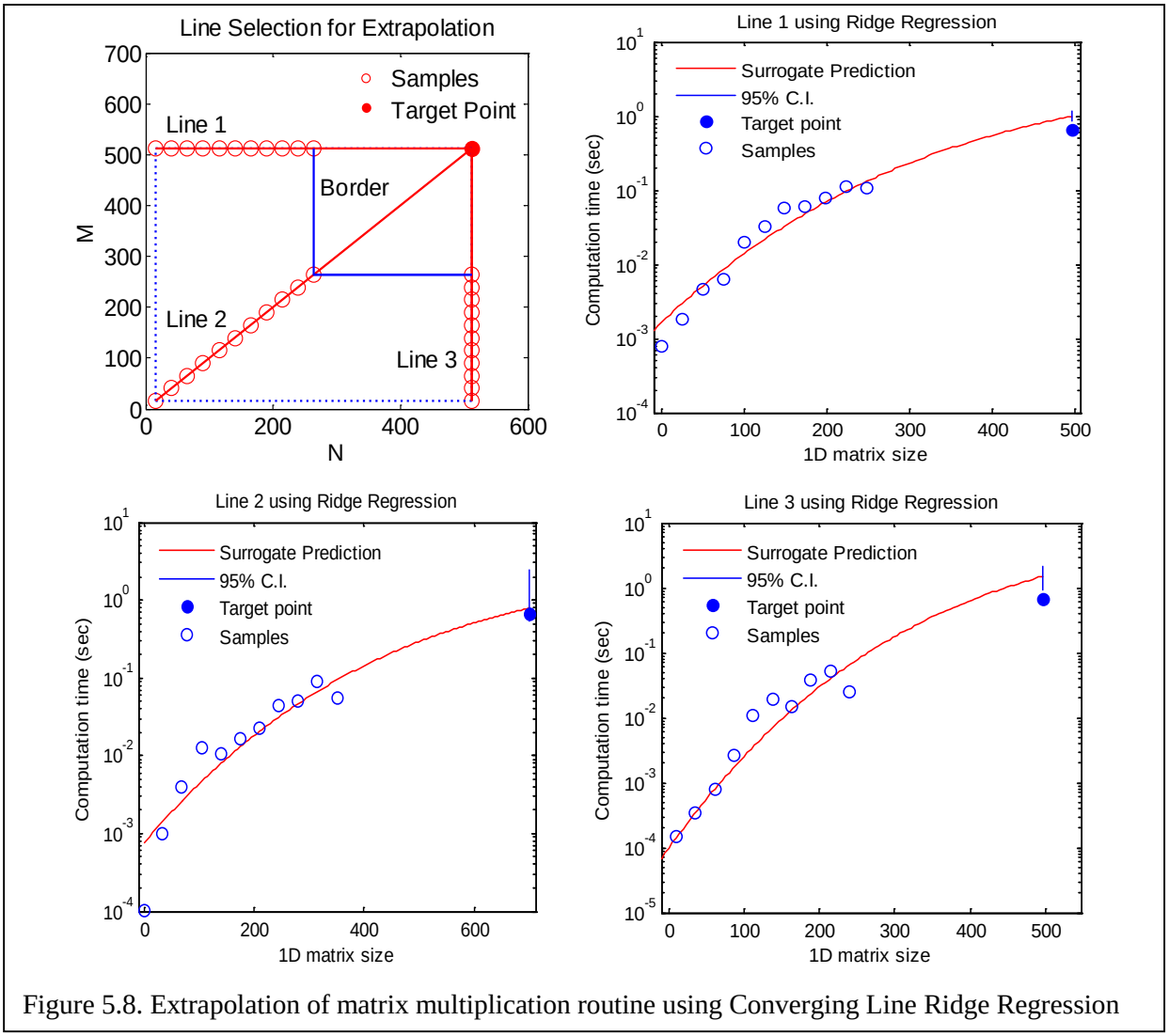


Figure 5.8. Extrapolation of matrix multiplication routine using Converging Line Ridge Regression

matrices. We could draw two guidelines: (1) Extrapolation method should be selected based on the density of the data and the level of noise. (2) With sparse data and large noise extrapolation based on standard regression was prone to over-fitting errors. To tackle the noisy data problem, we adopted ridge regression and proposed a parameter tuning procedure for better extrapolation as shown in Figure 5.8. We will incorporate this knowledge to the extrapolation problem of the microscale CFD simulation.

We also examined the Richardson extrapolation together with Microscale team to reduce discretization error of CFD simulations. We studied the case of 3D simulations that generate the drag coefficient of a spherical particle under the influence of shock. The standard Richardson extrapolation works well when the grids are refined such as element boundaries of coarser grids are still element boundaries for the finer grids. However, for 3D problems this will require an increase of 2^3=8 times the number of elements in each refinement. This is not going to be possible for some of our problems. So we examined what happened when grids are refined more slowly. In this case the movement of element boundaries generates noise. Richardson extrapolation was less robust with noise based on preliminary analysis. We reviewed the literature regarding to limitations of Richardson extrapolation. We are exploring the possibility of using instead polynomial regression with monotonicity constraints and statistical confidence intervals to estimate discretization error.

5.5 The Mesoscale shock tube experiments support

An undergraduate student, Justin Mathew, studies horizontal shock tube experiments conducted by Justin Wagner at Sandia National Labs to identify and understand uncertainty sources in the experiments. He has most recently been studying Wagner's particle curtain experiments in order to understand and characterize the methodology used to measure volume fraction as determined by x-ray images of the shock and particle curtain interaction. These characterizations can be used to validate code developed by the simulation team to model the particle curtain and shock interaction in the horizontal shock tube.

In addition to Wagner's experiments at Sandia, Justin has been working on characterizing input and output parameters related to the ASU mesoscale experiments conducted by Heather Zunino that would be useful to the simulation team modeling these experiments.

6. CMT-nek Code Development Team

6.1 Overview

CMT-nek demonstrated its capabilities in solving flow over exciting new configurations in 2015 which included Lagrangian tracking of 10^7 point particles in parallel for the first time in the history of nek5000. Difficulties in converging solutions to the full compressible Navier-Stokes equations led to a reformulation of the scheme that improved results and stability for prior demonstration problems. Debugging and testing continue, but conclusive results with publishable novelty for the fields of both discontinuous Galerkin methods and spectral element methods are pending as Year 3 begins. These formulation changes have delayed much of the progress promised in Year 2 (See Table 6.1 for realized progress), but will place them on a much firmer theoretical and mathematical foundation then they were before. The implementation of all formulation changes currently under review also fit cleanly into the CMT-nek codebase, which was committed to the nek5000 SVN repository at Argonne National Lab in November, 2015.

6.2 Viscous terms, dealiasing and reformulation

CMT-nek in 2014 aggressively filtered conserved variables every time step for stabilization. 2015 opened with the adaptation of the discontinuous Galerkin scheme (and quasi-linearization of the viscous fluxes) of Bassi & Rebay (1997) *J. Comp. Phys.* **131** (BR1 scheme) to the existing CMT-nek solver's formulation of all fluxes of conserved variables in strong form. This is consistent with how nek5000 handles first-order operators (like advection), and DG requires viscous diffusion to be expressed as a hierarchy of first-order equations. CMT-nek immediately provided good small-time solutions to some compressible flows with low viscosity, notably the initial growth of radial velocity in a viscous flow whose initial conditions correspond to those of a homentropic vortex (Colonius, Lele, and Moin (1991) *Journal of Fluid Mechanics* **230**) shown in Figure 6.2. However, the code failed to converge (preserve) steady viscous flows such as Couette flow with Chapman-Rubesin viscosity and could not produce the correct wall-normal velocity profile of compressible flow over an impulsively accelerated plate.

Since a productive visit to Argonne National Lab for the 2015 Nek5000 User/Developer Meeting, Paul Fischer's group began joining CCMT weekly meetings via teleconference. Our work on viscous terms and stabilization led us to reconsider dealiasing. Over the course of study and exploration on the basic building blocks of CFD (linear advection, diffusion and other problems), surface integrals were dealiased for the first time, and all operators were recast in weak form, eliminating the discontinuous surface integral and one integration by parts for the volume integral terms. This distinguishes CMT-nek from prior treatment of advection in nek5000, but weak form is the superior (and more conventional) choice for compressible flow.

Rarefaction flows have served as basic verification cases for the strong-form code throughout both fiscal years of CCMT's existence. In addition to periodic flow over the randomly located spheres (see Figure 6.1), a single sphere in a rectangular domain with outflow at the right end was subject

to a rarefaction with a tail Mach number of 0.3. The mesh and results of this case, run on Mustang at LANL, appear in Figure 6.3, with the drag coefficient (due to unsteady added-mass force) shown in Figure 6.4. Considering rarefaction without the sphere in a smaller two-dimensional domain, norm of the error in density (with respect to the output of a full numerical Riemann solver) in Figure 6.5 shows that the weak-form solver is significantly better (even without filtering) than the strong-form solver.

Mathematical formulation of DGSEM for compressible Navier-Stokes equations using the scheme and factorization of Bassi & Rebay (1997)	March 2015
Microscale demonstration: inviscid flow over 200 spheres in a periodic domain	May 2015
Adopted nek5000 conventions for thermodynamic properties, boundary conditions and state equations.	July 2015
Particle tracking implemented and demonstrated with tracers.	October 2015
Initial merge of CMT-nek into the official repository of nek5000 at ANL.	November 2015
CMT-nek reformulated in weak form of discontinuous Galerkin statement, replacing strong/ultra-weak form. Surface integrals dealiased separately.	December 2015
Stable convergence of homentropic vortex solutions of Euler equations.	January 2016
CMTbone for performance tuning	January 2016

Table 6.1. List of milestones reached in the second fiscal year, January 27th 2014 to January 27th 2015.

6.3 Particle tracking

During the second year, CMT-nek has been outfitted with particle tracking. The current capabilities include Lagrangian particles that are solely affected by the continuous phase, but not vice versa (one-way coupled). The particles are transported according to the three-step backward-difference scheme with forward extrapolation (BDF/EXT). For this computation, the particles are subjected to Stokes drag only, but it is easily extendable to other particle-fluid forcing.

One-way coupled Lagrangian point particle tracking can be broken down into two important steps – to locate and determine the computational coordinates of the point particle and to evaluate fluid values at the off-grid particle locations. In Q2Y2 we focused our efforts on developing an efficient and scalable algorithm for these two steps by leveraging existing capabilities in nek5000.

As the Lagrangian point particles are advected in the computational domain, they routinely move in and out of the pieces of the computational domain held by different MPI ranks. It is also important to understand that the MPI rank that holds the point particle properties, like co-ordinates, particle velocity, forces on the particle etc., can be different from the MPI rank that holds the piece of the computational domain which contains the coordinates of the point particle. The consequence of this scenario is that the process of evaluating flow properties at the point particle location has

to be done on a remote MPI rank and the interpolated values communicated back to the MPI rank which holds the properties of the particle. Clearly, the overheads associated with this can become crippling when large number of point particles are being tracked. Therefore, it was decided that the MPI rank that holds the computational domain, which contains the point particle location, should also hold the point particle properties. To achieve this, each MPI rank checks if the particle location is still within its domain after the end of a time step, when the particle locations have been updated. If it is, the computational coordinates are stored (among other quantities of interest). If the particle location is in another rank's domain, a separate routine (find_pts) is called to find which rank owns the particles and get the computational coordinates. Find_pts was already available in nek5000 code base. Once the identity of the remote MPI rank is known, an intrinsic nek5000 routine, crystal_tuple_transfer, which exchanges the particle properties between the necessary MPI ranks. At the end of this step all the particle properties are owned by the MPI ranks whose domain contains the point particle location. It should be noted that crystal_tuple_transfer has been shown to be scalable to millions of cores. Furthermore, without the existing framework of Nek5000, the potential scalability of CMT-nek would be rather difficult.

To compute the forces acting on the point particles, the fluid properties need to be interpolated at the off-grid particle locations. This past year we implemented a barycentric Lagrange interpolation routine that has been extremely reliable and highly scalable in practice. Even compared to the standard Lagrange interpolation, there is numerous evidence in literature that this form of polynomial interpolation is superior (Berrut and Trefethen (2004) *SIAM Review* **46(3)**). Also in our testing we have observed that that the barycentric Lagrange interpolation is faster that then the existing interpolation routines of neek5000.

Table 6.1 shows the performance of the CMT-nek particle tracking capability and compares it with the existing capabilities of nek5000. In the analysis we have selected 3 cases with 125000, 1000000 and 3375000 particles in a triply periodic 3D box with uniform flow. The particles were advected for 5000 time steps and average particle tracking time was computed. "old" refers to particle tracking using only the capabilities in nek5000 and "new" refers to the modifications implemented by CMT-nek which were described above. The table clearly shows that as we increase the number of point particles, CMT-nek algorithm is superior and substantially reduces the particle tracking time. As a demonstration of scalability of the point particle tracking feature, we simulated a case of point particles embedded in a rarefaction travelling in a long rectangular duct. Figure 6.6 shows density in the domain along with Lagrangian point particles. In this case there were 12 million point particles and the simulation was performed with 7680 MPI ranks on Mustang at LANL.

6.4 Third Year Plans

A summary of third year plans are listed in table 3. We will continue development of Lagrangian point particle tracking and implement 2-way coupling early Q2Y3. The implementation of weak form in CMT-nek will assure stable treatment of viscous diffusion, including artificial viscosity needed for shock capturing. The current development plan puts us on the path of performing the first simulation of the demonstration problem using CMT-nek by Q3Y3.

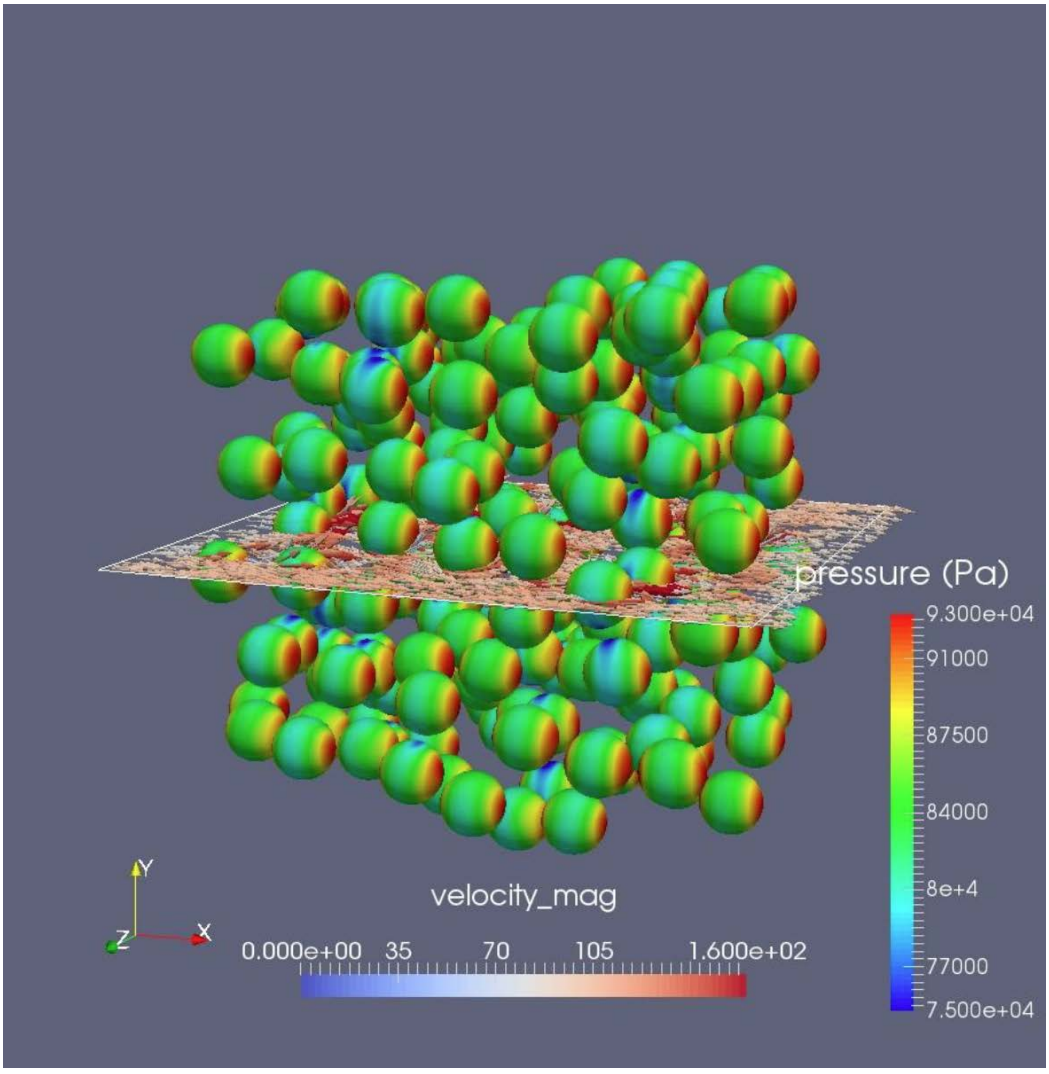


Figure 6.1. Pressure on the surfaces of 200 spheres in a periodic domain accelerated to Mach 0.3 in the x direction. Velocity vectors of the inviscid flow field are shown on the midplane in the y-direction. The domain has 67K elements and 23M grid points. The simulation took 11.1K hours on 182 nodes (4268 cores) on Mustang at LANL.

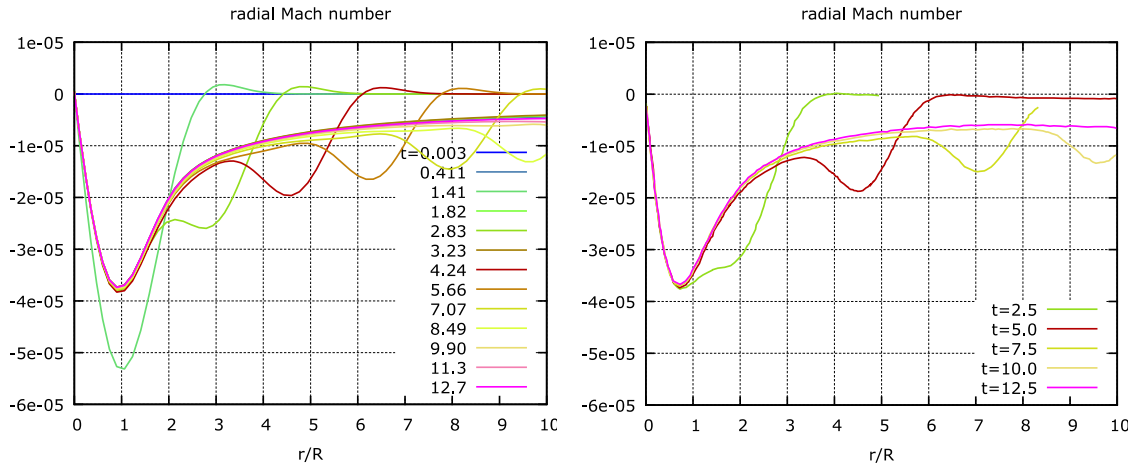


Figure 6.2. Radial velocity component normalized by sound speed in viscous flow of a vortex initialized to homentropic conditions. Maximum azimuthal velocity V_{max} is 0.5 times sound speed, $Re=V_{max}r_{core}/\nu=5000$. Time is normalized by acoustic time scale. Left: CMT-nek. Right: Colonius *et al* (1991).

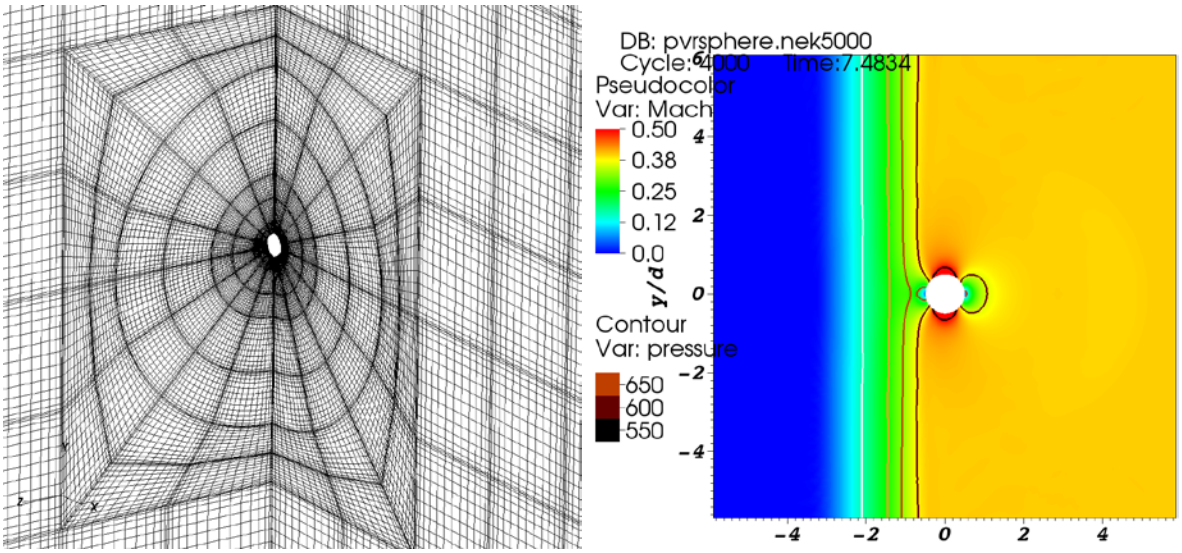


Figure 6.3. Rarefaction flow around a sphere in initially quiescent gas. 28K elements, polynomial order 14, 94M grid points, strong form code with filtering. Run on 440 MPI ranks on Mustang at LANL. Fan tail Mach number=0.3. Left: mesh detail; the domain measured 300x100x100 d_p. Right: Mach number pseudocolor with pressure contours in a midplane slice.

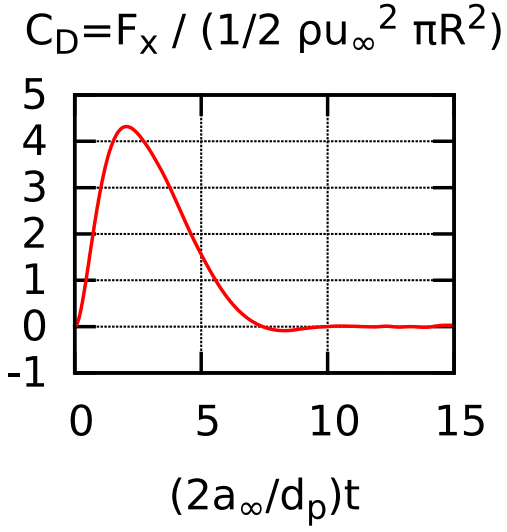


Figure 6.4. Drag coefficient of the sphere with a Mach 0.3 rarefaction moving over it in inviscid flow shown in Figure 6.3 vs time normalized by the acoustic timescale.

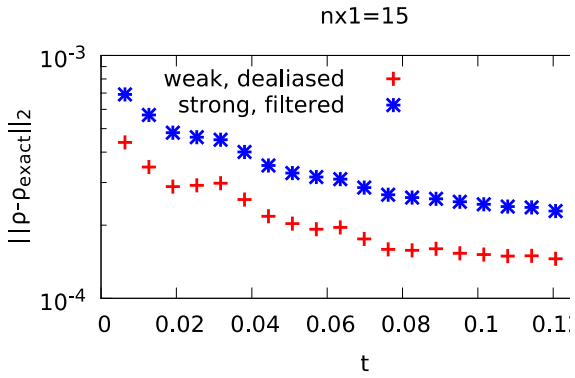


Figure 6.5. L_2 norm of error in density as a function of time for a rarefaction at (undisturbed) conditions identical to Figure 6.3, but in a smaller, two-dimensional domain. The curves on the bottom correspond to unfiltered results from CMT-nek reformulated in weak form.

Particles	**125,000**		**1,000,000**		**3,375,000**	
	Old	**New**	**Old**	**New**	**Old**	**New**
Flow solver	**32.37**	**32.08**	**32.41**	**32.24**	**32.57**	**32.48**
Particle tracking time	**45.54**	**21.46**	**207.91**	**34.82**	**641.29**	**77.58**
Get computational	**39.36**	**18.93**	**173.17**	**20.27**	**533.40**	**28.06**
Send to destination	**-**	**0.30**	**-**	**1.83**	**-**	**7.71**
Interpolation	**5.17**	**1.49**	**33.49**	**11.85**	**105.57**	**39.97**

Table 6.2: Performance of the CMT-nek particle tracking capability and comparison with the existing capabilities of nek5000. “old” refers to particle tracking using only the capabilities in nek5000 and “new” refers to the modifications implemented by CMT-nek which were described above.

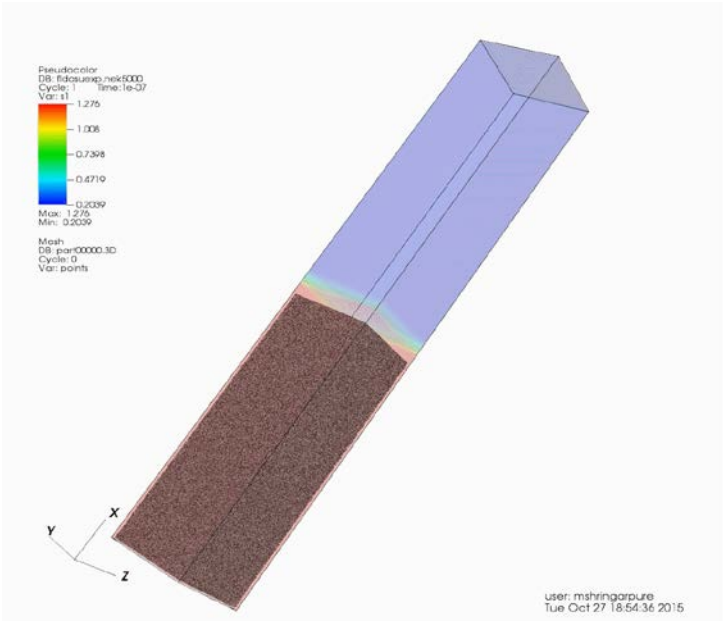


Figure 6.6. Particles and density volume rendering for rarefaction-driven particles at Mach 0.3. The domain has 12 million point particles, and the simulation was performed with 7680 MPI ranks on Mustang at LANL.

Capability	Test cases	Time frame
Compressible Navier-Stokes	• Viscous flow over a sphere, low Mach, low Re • Compressible isotropic turbulence	May 2016
Lagrangian point particles	• 2-way coupling	June 2016
Shock capturing	• SOD problem • Sedov problem • Shock propagating over array of spheres (microscale simulations for CCMT)	July 2016
Dense to Dilute volume fractions	• Moving shock over curtain of particles and cylindrical explosive dispersal of particles (Macroscale simulation for CCMT)	November 2016

Table 6.3. List of capabilities that would be added to CMT-nek in the third fiscal year. Also shown are the projected completion dates and list of test cases to be used to verify the code.

7. CS Team

7.1 Overview

The primary goal of the research of our CS team has been to improve performance, power consumption and thermal characteristics of the CMT-nek application, and in the process find new strategies and optimization ideas that can be applied to other applications as well. The three main problems we focused on last year are hardware/software co-optimization of the derivative computation kernel using genetic algorithm based autotuning, hybrid (CPU + GPU) implementation of the derivative kernel of CMT-nek and efforts to GPUizing CMT-nek.

7.2 Autotuning

7.2.1 Background

Over the first year, we focused our efforts on improving the performance and power consumption characteristics of the spectral element solver which is the most compute intensive kernel of Nek5000 and is being reused in CMT-nek. The fundamental algorithm of this kernel is small matrix-matrix multiply that is used to numerically compute functional derivatives. We used a number of optimization techniques on CPU and GPU to achieve improved results compared to the original nek5000. On CPU, the main optimization technique used was auto tuning and we tested our results on IBM Blue Gene/Q, 64-core AMD Opteron 6378 and AMD-Fusion. We further developed a genetic algorithm based driver to quickly determine a near optimal version of the derivatives.

Work done last year: We ran our genetic algorithm based autotuning strategy on Cab (Intel SandyBridge platform at LLNL) under different power bounds. We also extended our autotuning framework to include cache parameters which enabled us to perform hardware-software co-optimization. When combined with the software parameters, this resulted in a tremendous increase in the parameter search space.

7.2.2 Intel Xeon Platform

We used Intel's Running Average Power Limit (RAPL) to measure the average power consumption as well as to set an upper bound on the average power consumption over an interval of time. Perhaps the most interesting and unique study on this Intel platform is the behavior of the code variants under different power bounds. Without a power bound, the native implementation consumes about 96W. We present here the results for 10 power bounds ranging between 50W and 95W. The DRAM is not power bounded in these experiments, because this feature is not supported on Cab. Figure 7.1 shows how runtime drops as power bound is relaxed. The graph compares the performance of the native implementation versus that of the best version found by using the GA algorithm for each power bound. The optimal algorithm was different for each power bound. Thus, the nature of the optimal algorithm in terms of loop permutation and unroll factors depend on the matrix size as well as the power bound.

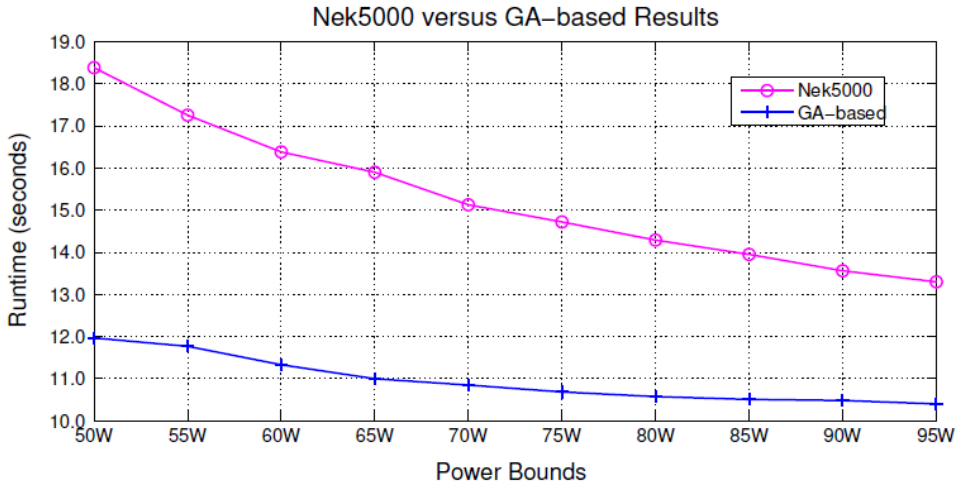


Figure 7.1. How runtime changes for ∂U/∂r calculations using the native implementation in Nek5000 and optimal implementations found using GA driven autotuning for each CPU power bound. The best versions for each power bound have different characteristics for loop permutation and unroll factors.

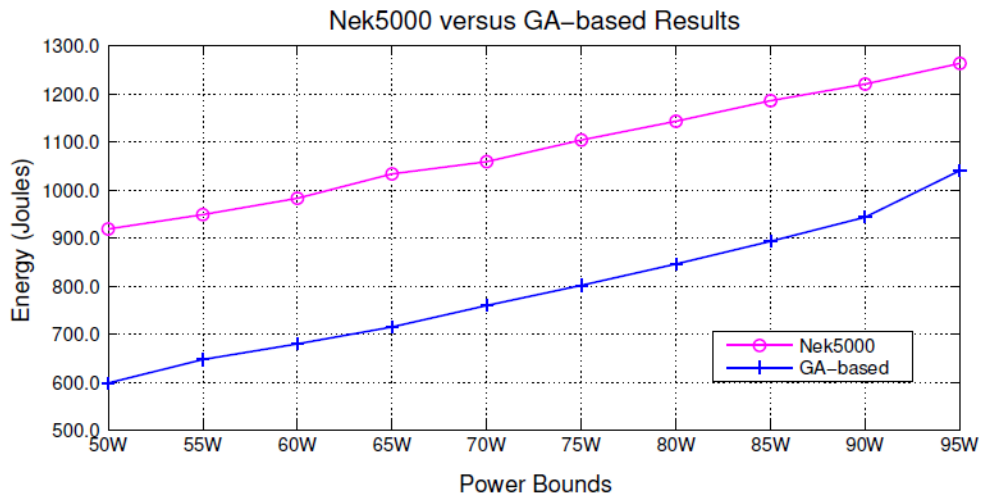


Figure 7.2. Energy consumption increases with power bounds for ∂U/∂r calculations using the native implementation in Nek5000 and optimal implementations found using GA driven autotuning for each CPU power bound. Energy consumption in minimum for the lowest power bound and increases as power bound is increased.

Figure 7.2 shows how energy consumption increases as power bound is relaxed. From this graph we find that operations at lower power bounds are more energy efficient. Thus the performance gain obtained by using more power does not compensate for the excess power consumed in the energy equation represented by the product of power consumption and execution time.

7.2.3 Gem5 Platform

We used gem5, the well-known micro-architectural simulator, and McPAT, the power estimator, to evaluate different cache configurations for energy and performance trade-offs for out-of-order x86 cores at the micro-architectural level for derivative computation. The cache parameters considered and their values are:

- Cache parameters: For L1 cache, the parameters considered were cache size, line size, set associativity. L1 cache is most relevant for this problem since it involves small matrix multiplications.
- Parameter Values:
  - Cache size: 2KB, 8KB, 16KB
  - Cache line size: 64B, 128B, 256B
  - Set Associativity: 2, 4, 8

Certain combinations of the above parameter values are rendered inadmissible by gem5 and/or McPAT. The table in Figure 7.3 gives the admissible combination of values we used for our experiments. Clock speed was 1GHz. We now present results from exhaustive versus genetic algorithm based search.

Combination Index	Cache Size (KB)	Line Size (B)	Set Associativity
0	2	64	2
1	8	64	2
2	8	64	4
3	8	128	2
4	8	128	4
5	16	64	2
6	16	128	2
7	16	128	4
8	16	128	8
9	16	256	2

Figure 7.3. Table of admissible combinations of cache parameters used in our experiments.

7.2.4 Exhaustive search versus genetic algorithm driven search

We ran exhaustive search for processing an element represented by 16x16x16 grid of function values (N = 16) on gem5 to get the best derivative computing codes and the best setting for cache configuration for a subroutine that computes the partial derivatives of the function. The total number of code variants analyzed was 4956 for $\partial U/\partial r$ and $\partial U/\partial t$, and 4559 for $\partial U/\partial s$. Each code variant was tried on 10 different cache configurations (Figure 7.3).

Figure 7.4 shows a comparison of run times for native nek5000 implementation, compared to the best versions we get using exhaustive autotuning and GA based autotuning. Performance can be improved by over 40% compared to the native implementation, using a combination of code transformations and cache configuration. Of particular interest are the GA based autotuning results. The variants identified as the best for each derivative by GA based driver were very close to the optimal variants and these were found after analyzing only about 0.25% of the total variants analyzed by exhaustive search. The configuration used for the native nek5000 implementation is shown in Figure 7.5.

Derivatives	NEK5000 Time (s)	Exhaustive Autotuning			GA based Autotuning		
		Time(s)	Variants	%impro-vement	Time(s)	Variants	%impro-vement
dudt	3.61e-04	2.00e-04	49560	44.6	2.03e-04	120	43.76
dudr	9.39e-05	8.99e-05	49560	4.4	9.00e-05	120	4.44
duds	1.28e-04	9.50e-05	45590	25.8	9.60e-05	120	25.00

Figure 7.4. Comparison of compute times in seconds, between NEK5000, the best variant using exhaustive autotuning and the best variant found using genetic algorithm based driver.

Characteristic	dudr	duds	dudt
Loop permutation	123	1234	123
Loop unroll factors	1, 1, 16	1, 1, 1, 16	1, 1, 16
Cache size	16KB	16KB	16KB
Line size	64B	64B	64B
Ways of associativity	4	4	4

Figure 7.5. Table of code and cache characteristics for nek5000 code.

Characteristic	dudr	duds	dudt
Loop permutation	1234	1234	1234
Loop unroll factors	16, 1, 1, 1	16, 1, 1, 1	2, 1, 2, 2
Cache size	8KB	16KB	8KB
Line size	128B	256B	64B
Ways of associativity	4	2	2

Figure 7.6. Table of code and cache characteristics for the best performing code using exhaustive autotuning.

Cache configuration of Figure 7.5 was matched up with the existing configuration on HiPerGator, the AMD Opteron platform at UF. Nek5000 uses 4loop-fused implementation for ∂U/∂r and ∂U/∂t, hence there are only three loops and three unroll factors, for these derivatives, with the innermost loop being completely unrolled. Figures 7.6 and 7.7 show the code transformation and cache

configuration parameters used to obtain the best variant for the three derivatives using exhaustive autotuning and GA based autotuning, respectively.

Characteristic	dudr	duds	dudt
Loop permutation	1234	1234	2134
Loop unroll factors	16, 1, 1, 4	16, 1, 2, 1	1, 2, 1, 1
Cache size	8KB	16KB	16KB
Line size	128B	128B	128B
Ways of associativity	4	4	2

Figure 7.7. Table of code and cache characteristics for the best performing code identified by genetic algorithms.

Figure 7.8 shows how changing cache configuration helps in improving performance. We get an additional improvement of 2.5%, 1.1% and 25.7% upon using a different cache configuration.

Figure 7.9 shows a comparison of energy consumption for native nek5000 implementation, compared to the best versions we get using exhaustive autotuning and GA based autotuning. As in the case of performance improvement we see that energy consumption can be improved by over 40% compared to the native implementation.

Derivatives	NEK5000 Time (s)	Exhaustive Autotuning		
		Time1 (s)	Time2 (s)	%improvement
dudt	3.61e-04	2.05e-04	2.00e-04	2.5
dudr	9.39e-05	9.09e-05	8.99e-05	1.1
duds	1.28e-04	1.28e-04	9.50e-05	25.7

Figure 7.8. Comparison of compute times in seconds, between nek5000, the best variant with the same cache configuration as used for running nek5000 (Time1), and the best variant taking into consideration all cache configurations (Time2). A change in cache configuration is most effective for $\partial U/\partial s$ computation, with a reduction in runtime by 25%.

Derivatives	NEK5000 Energy (J)	Exhaustive Autotuning			GA based Autotuning		
		Energy (J)	Variants	%improvement	Energy(J)	Variants	%improvement
dudt	0.0134	0.00723	49560	46.00	0.00737	120	45.00
dudr	0.0034	0.00328	49560	3.44	0.00329	120	3.26
duds	0.0046	0.00350	45590	23.90	0.00353	120	23.26

Figure 7.9. Comparison of energy values in Joules.

Figures 7.10 and 7.11 show the energy performance characteristics of some variants that are close to the optimal variant for the derivatives. Each dot in these graphs is a variant representing a unique combination of code transformation parameters for N=16 and cache configuration parameters in gem5. Though the best variant (which was identified using exhaustive approach) is indeed the most energy efficient variant for ∂U/∂r (and so also for ∂U/∂t), that is not the case for ∂U/∂s. In fact, for the latter, there are three variants that consume lower energy than the best performing variant as shown in Figure 7.10. This shows that the best performing code does not necessarily consume the least energy.

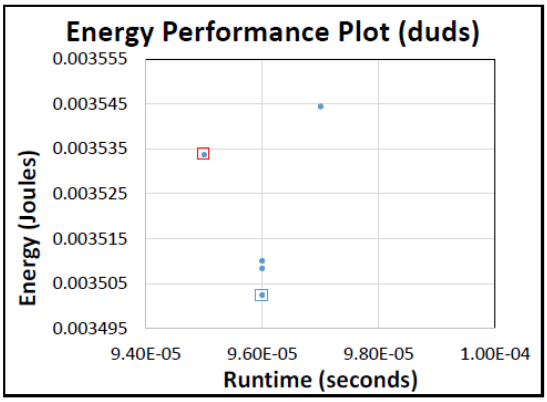


Figure 7.10. Energy performance plot for the different variants close to the optimal variant for ∂U/∂s. The performance optimal variant is enclosed by a red box, whereas the most energy efficient version is enclosed in a blue box.

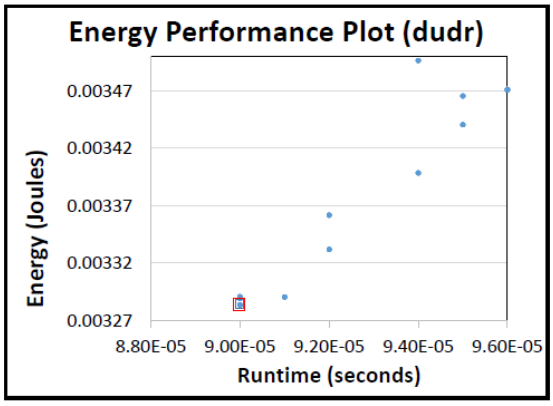


Figure 7.11. Energy performance plot for the different variants close to the optimal variant for ∂U/∂r. The same variant is both performance and energy optimal and is enclosed in red and blue boxes. The plot for ∂U/∂t is also similar.

7.3 Hybrid (CPU + GPU) implementation of derivative kernels of CMT-nek

We used optimized CPU and GPU implementations for the derivative computation kernel and then distributed these computations to both CPU and GPU, studying the overall performance and energy consumption characteristics. The CPU implementation includes vectorization of code as well as using platform specific optimizations of using loop permutation and unroll in the code. We developed empirical models for CPU and GPU performance, power and energy consumption. We derive load balancing techniques for dividing up the work on heterogeneous cores on an HMP based on the empirical models of CPU and GPU cores. These techniques may be used to obtain either a performance optimal or an energy optimal implementation, depending upon the optimization objectives. We find that the computational workload assignment, in general, depends on whether the objective is to optimize for performance or for energy.

Our experiments were run on an AMD Opteron 6168 host with 12 CPU cores, running at 1.9GHz clock frequency. The processor is connected to a Tesla K20c GPU (Kepler architecture) via a PCI Express slot. We use one core as the GPU host and the remaining 11 cores are used to study performance scaling, power and energy usage of the application.

We first begin with our modeling results. We model the GPU and CPU characteristics such as performance, power and energy consumption, based on two parameters: matrix size, N, and the number of elements Y. The modeling helps us gain insight into the relationship of performance characteristics of small matrix multiplications with respect to matrix size and the number of elements. The models helps us to understand the load balancing problem on CPU and GPU, and provides us with a good starting point to achieve optimum performance and energy consumption. The alternate method for load balancing using autotuning or binary search takes up significant machine time as the program has to be run several times to find the optimum load distribution to CPU and GPU. The models are also useful to theoretically analyze the characteristics of CPU-GPU configurations that are not available to us for direct experimentation.

7.3.1 Modeling on GPU

The two components of GPU performance are computation time and communication time. Let T_{comp} and T_{comm} represent the computation and communication times respectively, and T_{GPU} represent the total runtime taken by the derivative computation kernel to run on GPU. Using a linear least squares fit on a plot of logarithm of the data, its slope and the y- intercept gives us the exponent of the term *N* (matrix size) and its coefficient respectively in our model. We similarly determine the exponent and coefficient of the term *Y* (number of elements).

Using multiple regression, we obtain combined models for Tcomp and Tcomm based on both N and Y.

$$Tcomp = 7.17 \times 10^{11} \times N^{3.76} \times Y$$
$$Tcomm = 6.14 \times 10^{9} \times N^{3} \times Y$$
$$Tgpu = Tcomp + Tcomm$$

To check how well the model fits the data, we analyzed R-squared value and root-mean-square error (RMSE). R-squared is a statistical measure of how close the data are to the fitted regression line. Possible value of R-squared is between 0 and 1, where 0 is the worst case in which no data is matched by the fitted line. For the equation modeling Tcomp, R-squared is 0.9974 and RMSE is 0.1311, and for that modeling Tcomm, R-squared is 0.9934 and RMSE is 0.1935.

Using a similar strategy we model GPU power and energy for the derivative computation kernel on our platform and these are given by the following equations obtained after performing multiple regression analysis.

$$Pgpu = 162.24 \times N^{-0.1}$$

$$Egpu = 9.28 \times 10^{-7} \times N^3 \times Y$$

Energy should always be power times time. Again, these are the best fits based on empirical data and may not follow the above requirement precisely. For the power equation R-squared is 0.5146, though RMSE is 0.03929. Power consumption does not depend on the number of elements processed, and overall power consumption decreases very slightly as larger matrices are used. This happens because of two factors namely, the number of memory transactions per unit of data transferred decreases and the computation to data transfer ratio improves with increasing N. For the energy equation, R-squared is 0.9941, whereas RMSE is 0.182. The energy consumption equation for EGPU is very similar to the equation for computing TGPU, which shows that energy consumption is proportional to runtime.

When multiple GPUs are present, the workload could be distributed uniformly and for *g* GPUs we could approximate the performance trend to

$$Tggpu = \frac{1}{g} Tgpu$$

It is possible to achieve the 1/g factor since the GPUs need not communicate with each other to solve this problem.

7.3.2 Modeling on CPU

CPU runtime Tcpu, power Pcpu and energy Ecpu can be similarly modeled as follows:

$$Tcpu = 1.02 \times 10^{-9} \times N^{4.4} \times Y$$
$$Pcpu = Pmemory + Pcore$$
$$Pmemory = 5.95$$
$$Pcore = 12.21 \times N^{0.3}$$
$$Ecpu = 2.51 \times 10^{-8} \times N^{4.5} \times Y$$

When multiple CPUs are present, the workload can be distributed uniformly and we model the runtime using *p* CPUs as follows.

$$Tpcpu = \frac{1}{p} Tcpu$$

7.3.3 Modeling on hybrid platform

In this section we model performance of the derivative computation kernel on hybrid multiCPU, multiGPU systems based on the models we already developed so far. The model can be used for load balancing to determine optimal distribution of workload between the CPU cores and the GPUs and determine the minimum runtime of the combined system. Since the work done by CPU core overlaps with that on the GPU, the total execution time is best modeled as

$$Teff = (Tcpu, Tgpu)$$

Thus, *Teff* is minimized when *Tcpu* = *Tgpu*. This condition can be used to theoretically obtain the minimum runtime and also to determine the *Y* 's in equations for *Tcpu* and *Tgpu*. For example,

suppose ycpu and ygpu are the number of spectral elements processed by the CPU and the GPU respectively.

Then if the total number of elements (ycpu + ygpu) and the matrix size are given, one can easily solve for ycpu and ygpu from the condition *Tcpu* = *Tgpu*. Figure 7.12 shows for different matrix sizes, *N x N x N*, the execution time (in seconds) on hybrid architectures using the empirical model

pCPU-gGPU	Matrix size ($N \times N \times N$)	Model		Optimum		Deviation of time(%)
		$y_{cpu}\%$	$time(s)$	$y_{cpu}\%$	$time(s)$	
1-1	$8 \times 8 \times 8$	24.02	0.0214	20.44	0.0218	1.83
	$9 \times 9 \times 9$	21.02	0.0314	18.44	0.0321	2.18
	$10 \times 10 \times 10$	18.58	0.0442	20.53	0.0430	2.79
	$11 \times 11 \times 11$	16.59	0.0601	17.25	0.059	1.86
	$12 \times 12 \times 12$	14.92	0.0794	15	0.0782	1.53
	$14 \times 14 \times 14$	12.32	0.1293	10.62	0.1349	4.15
	$16 \times 16 \times 16$	10.41	0.1968	9.75	0.2072	5.02
5-1	$8 \times 8 \times 8$	61.95	0.011	56.25	0.012	8.33
	$9 \times 9 \times 9$	57.77	0.017	53.12	0.0186	8.60
	$10 \times 10 \times 10$	53.97	0.0254	56.25	0.0239	6.28
	$11 \times 11 \times 11$	50.50	0.0363	51.56	0.0348	4.31
	$12 \times 12 \times 12$	47.34	0.0499	46.87	0.0497	0.40
	$14 \times 14 \times 14$	41.85	0.0869	37.49	0.0942	7.75
	$16 \times 16 \times 16$	37.27	0.1393	33.97	0.1524	8.60

Figure 7.12. Comparison of the runtime obtained for hybrid implementation of derivative computation for 10000 spectral elements with empirical model and optimum experimental time. Ycpu% is the percentage of load on CPU. Model predicted time is within 10% of actual time.

and the optimum obtained using actual experiments for 10000 spectral elements, along with a percentage of the number of elements y_{cpu} processed by the CPU in each case. The last column in this table gives the deviation of the actual runtime from the predicted runtime.

For a p-core CPU and g-GPU system,

$$Teff = \max(Tpcpu, Tggpu)$$

to which we can similarly apply the minimization condition *Tpcpu* = *Tggpu* to get the minimum runtime as well as the distribution of workload among the processing units.

Thus, the condition for minimizing runtime is that all the units must finish processing their load at about the same time. For minimizing energy consumption, on the other hand, we must minimize the sum of energy consumption of each individual unit. This is explained with an example from our experiments in what follows. One interesting observation there is that energy consumption of a hybrid system is not directly proportional to runtime.

7.3.4 CPU + GPU experiments

7.3.4.1 Performance scaling

Figure 7.13 shows how performance of the kernel changes with increasing number of processing elements used to process 10000 spectral elements.

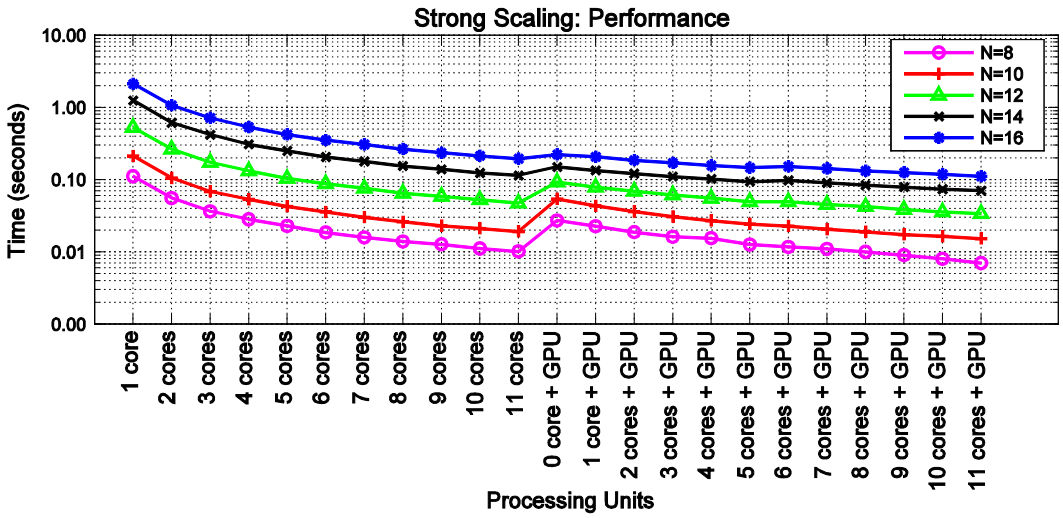


Figure 7.13. Performance as 10000 spectral elements are processed by an increasing number of processing units, for different matrix sizes. Runtime decreases by 20%-90% as a GPU core is added to any given number of CPU cores.

7.3.4.2 Power

Figure 7.14 shows the performance per Watt in terms of GFLOPS/Watt which is represented in log scale on the y-axis of the graph. Performance per Watt improves by an order of magnitude when a GPU is used, the highest performance per Watt is obtained when the GPU is used all by itself.

7.3.4.3 Energy

Figure 7.15 gives the energy consumed by the CPU-only and the CPU+GPU configurations.

7.3.4.4 Performance versus energy tradeoff

Figure 7.16 gives the performance and energy consumption of the various hybrid configurations we experimented with. The energy performance tradeoff can also be cast as an energy optimization problem under timing constraints which may be formulated as follows:

Minimize $Epcpu + Eggpu$ Ycpu, Ygpu

subject to

$$Tpcpu \leq T1$$
$$Tggpu \leq T1$$
$$Ycpu + Ygpu = Y1$$
$$g \leq G1$$
$$p \leq P1$$

Where p and g are the number of CPU cores and GPU devices respectively, T1 is the target runtime, Y1 is the total number of elements, G1 is the number of devices accessible from the node and P1 is the number of cores on the node. Note that we do not constrain the various processing units to finish processing at the same time as we did to obtain optimal performance.

To solve for this problem we see that the optimum energy consumption happens when the GPU processes as much load as is possible within the timing constraint with the remaining load being processed by the CPU cores. This is further explained in our hybrid paper submitted to SUSCOM.

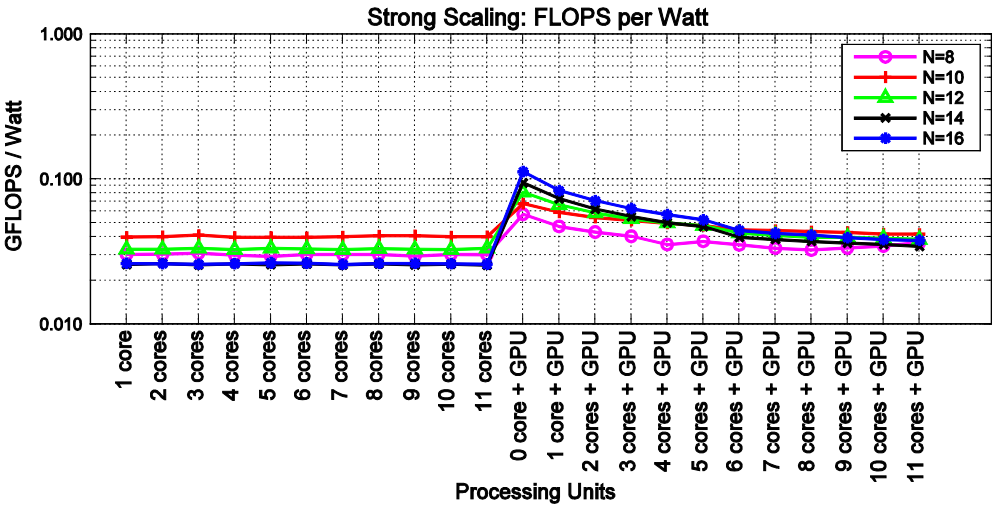


Figure 7.14. Performance per Watt for different combination of processing units and matrix sizes. Processing the smaller matrices on CPU is more efficient whereas the GPU is more efficient for the bigger matrices. As more cores are added with the GPU, the differences in efficiency of CPU and GPU based on matrix size cancels out and the lines tend to converge.

7.3.5 Ratio of CPU to GPU loads

Figure 7.17 gives the ratio of the number of spectral elements processed GPU and CPU cores for optimal performance. Compared to a single CPU core, GPU with all its cores is able to process about 9 times more load. In this case, we have included the communication time between CPU and GPU towards the time taken by GPU to process the loads. If we exclude the communication time, then a GPU can process 1000 time more load than a single CPU core.

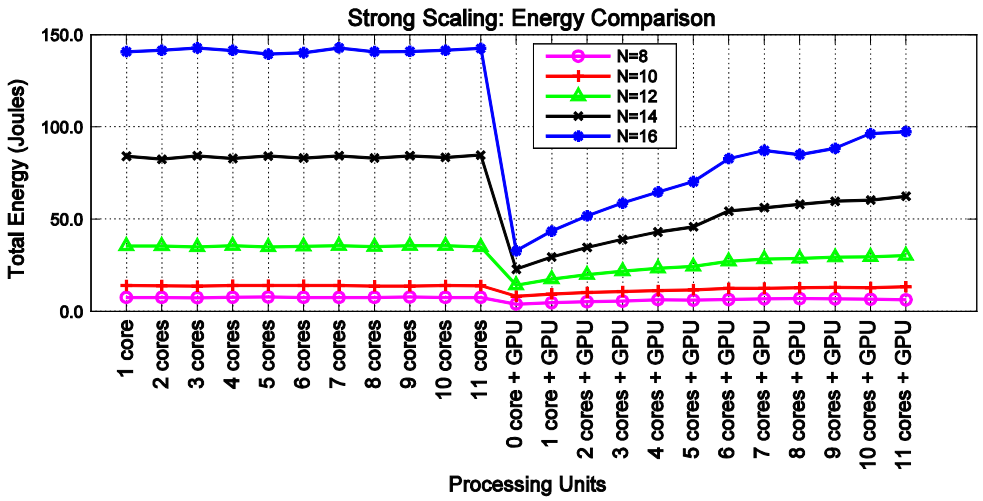


Figure 7.15. Energy Consumption. Total energy remains unaffected in CPU-only configurations since increase is power is offset by a proportional decrease in runtime as more cores are used. With GPU, energy consumption increases as more CPU cores are used. Even then, energy consumption for 11 CPU cores is about 1.5 times that for 11 cores + GPU.

7.3.6 GPUizing CMT-nek

Given the fact that the very next generation of supercomputers are being designed to have GPU accelerators, we want to support CMT-nek on GPUs. Towards this goal, we defined the following steps:

1) Development of CMT-bone that captures the key data structures and operations of CMT-nek.
2) GPUizing CMT-bone.
3) GPUizing CMT-nek.

We spent the end of last year identifying the data structures and operations we want in CMT-bone. The key data structures identified are volume and surface data structures along with point particles. The key operations identified are

1) point computations of variables such as mass, momentum, energy, pressure, density;
2) exchange of surface data with neighbors;
3) computation of partial spatial derivatives of fluxes;
4) particle initialization, distribution and tracking.

This is work in progress, and will take at least the first half of 2016.

7.4 Third Year Plans

Below is a list of tasks that the CS team expects to complete in the coming year.

1) CMT-bone development and validation, specially representing weak form of discontinuous Galerkin methods as corresponding changes are made to CMT-nek.
2) Continuing our work on performance and energy optimization, we will apply our autotuning technique on the new CMT-bone with weak formulation.
3) We will work of a GPU version of the CMT-bone.
4) We will work on power modeling and research on thermal aware computing.

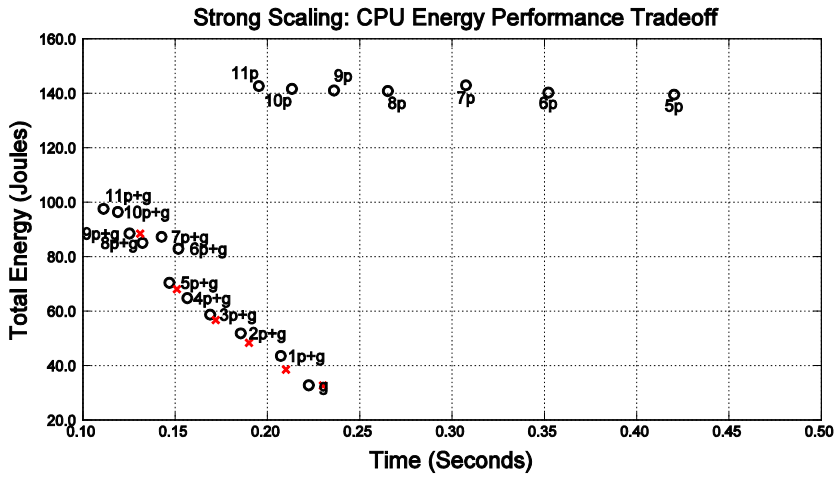


Figure 7.16. Energy-performance tradeoff. 'p' stands for CPU core and 'g' for GPU. The results are for matrix size 16 x 16 x 16 and number of spectral elements is 10,000. The most energy efficient implementation is that on a GPU while the implementation with the least runtime is one on 11 CPU cores and GPU. The CPU-only implementations consume large amount of energy. Red crosses represent energy optimal implementations.

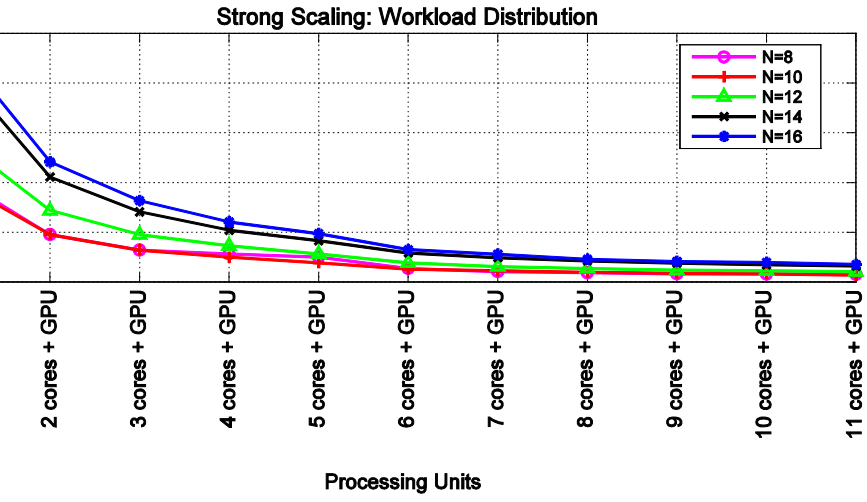


Figure 7.17. Ratio of the number of spectral elements processed by the GPU versus CPU cores for optimal overall performance. For the matrix size 16 x 16 x 16, GPU processes about 9 times more spectral elements to balance the load.

8. Exascale Team

8.1 Overview

The Exascale research focuses on enabling rapid design-space exploration of design strategies, optimizations, and parallelizations on extreme-scale systems up to Exascale. Such exploration supports the project by providing insight into how to optimize CMT-nek for potential candidate architectures before those architectures exist.

The Exacale research currently consists of four main thrusts, which are described in the following sections. Section 8.2 discusses Behavioral Emulation, which is our unique approach to abstracting away low-level details to improve scalability, while retaining sufficient accuracy for design-space exploration. We will describe the experiments that we have performed, using the developed Behavioral Emulation methods running on an experimental platform. We will also report on our progress in the development of a prototype "production" platform. Section 8.3 discusses uncertainty quantification to help establish an understanding of the reliability of behavioral emulation. Section 8.4 discusses FPGA-acceleration techniques for a range of design-space exploration options. Section 8.5 presents a network performance characterization to gain insight into the communication characteristics of CMT-nek, which in turn will provide insight into how to optimize the communication aspects of CMT-nek for future systems.

8.2 Behavioral Emulation (BE) Methods and Platforms

The goal of Behavioral Emulation (BE) is to permit design-space exploration (DSE) for future Exascale applications and architectures by the use of simplified, component-based models which mimic much of the behavior of those applications and architectures. The focus of this term, with regards to the conceptual underpinnings of Behavioral Emulation, was to ensure that BE-style models are able to accurately predict the performance of assorted test applications and architectures, where the test cases are sufficiently complex as to be non-trivial, but at this point not near Exascale.

The principal experiment of this term involved a test application based on a simplified version of CMT-bone. This application was simulated to run on the Cab system, and was also actually run on the Cab system for a variety of different system sizes and parameters. From a comparison of the simulated and measured execution time of this application/architecture pair (see Figure 8.1), we can confirm that BE-styled models are sufficiently accurate to continue our current research plans, and can be presently used for limited design-space exploration. In the following sections, we outline our plans to perform some directly-contributive DSE to assist the code development and computer science teams.

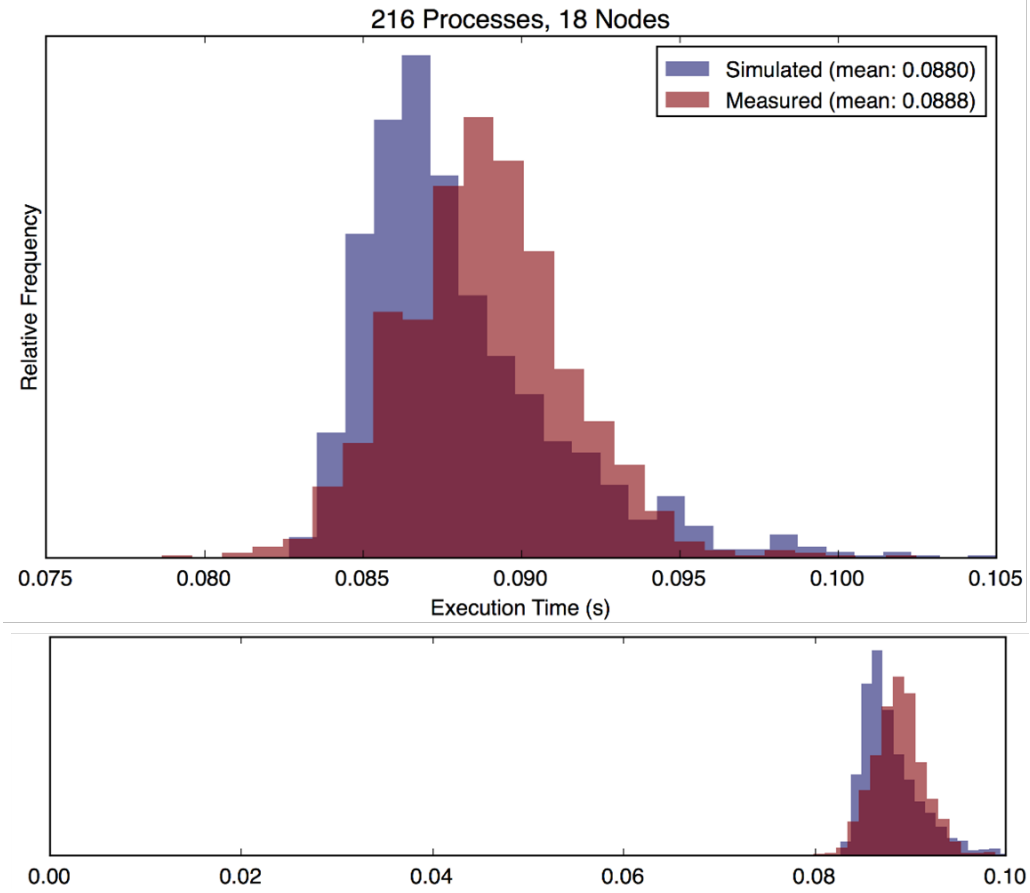


Figure 8.1: Comparison of histograms of Simulated (blue) and Measured (red) execution time for 1000 runs of previously-described test application on Cab, running on 18 nodes. The reasons for the differences in distribution are known and described in past presentations.

A summary of the progress within this task is given below, where those items tagged with "methods" are concerned with conceptual issues associated with BE, and those tagged with "platforms" are concerned, ultimately, with the creation of a production simulator which employs BE.

Methods:

- By comparing the measured and simulated execution time of a test application (which represents a very simplified variant of the core communication and computation patterns CMT-nek) on several nodes of the Cab computer at LLNL, we can confirm that behavioral models are effective in capturing the important properties of components which are more complex than a single device (which had been the focus of previous studies).
- Started exploring the use of probabilistic (many-run Monte Carlo) simulation to better and analyze the behavior of current and future generation systems. By performing Uncertainty Quantification, we should be able to both improve the accuracy of our models for

current/future systems, and provide an estimate of the certainty of our predictions for future systems.

- Began evaluating and exploring the use of different methods of accelerating simulations. These methods are varied, as is our certainty of their plausibility as ways to speed up simulation, but there are broadly two categories: those which are intended to speed up any computer simulation, and those which only work to speed up a set of related candidate simulations. The latter category, some of the methods of which are described in Section 8.4, are intended to allow the simulation of many similar application / architecture varieties in a time close to that of a single simulation.

Platforms:

- Completed development of an experimental simulation platform which conforms completely to the basic specifications of BE models and simulation notions. This platform was used in the Cab simulations to evaluate the efficacy of BE, and is also used as a reference in the creation of the production simulation platform.
- Completed roughly half of the production simulation platform, which is based on SST. The speed and established code base of SST should allow us to achieve our modeling goals, and initial testing of the performance of this simulation platform shows that its capability will probably allow us to reach the required simulation scale.

Plans for 2016:

Methods: Use BE methods to decide among a set of candidate implementations for a particular piece of the future CMT-nek codebase (this piece concerns the tracking and load-balancing of particles). This will be the first full-scale test of BE methods, in that: (1) the simulated applications will be derived from real candidate code, and (2) the simulated architectures will be large. This task will show the value and capability of BE in application design-space exploration for future systems, and will require considerable collaboration with the CS and Physics teams.

Methods: Evaluate the true speedup, by simulating large systems and applications, achieved by the various acceleration methods which, in this past year, were proposed and given cursory examination.

Platforms: Complete the development of the simulation platform built upon SST. Additionally, perform various tests to characterize its performance, including tests to see how well it scales when run on several nodes (simulation nodes, not simulated nodes).

8.3 Uncertainty Quantification (UQ) for Behavioral Emulation

We contend that quantification of errors and uncertainties is required to establish reliable/credible predictions with our BE methods. Past reports have described our procedure for the validation & verification (V&V) of BE models targeting existing architectures. When we predict the performance of some test application on an existing architecture our results can be trusted because

we have the ability to compare the predicted result with actual data collected from the target. The same cannot be said for BE prediction of purely notional (future) architectures. In this thrust we aim to collaborate with the UB team to enhance our previous V&V efforts to quantify the effect that uncertain inputs and model imperfection have on BE accuracy (e.g., determine variance and confidence intervals for simulation output, analyze output sensitivity to variance in simulation inputs, quantify uncertainty when using calibrated BE models for prediction).

At its core, BE is a discrete-event simulation of various coarse-grained compute and communication events – compute being the modeling for a collection of local operations executed on a specified target architecture and communication the modeling of messages traveling through a target network from one point to another. Along with computation and communication events, implicit wait and queue events are used as slack variables to reconcile thread-level clocks after a communication event between threads – implicit wait when a thread is ready to receive before messages arrive and implicit queue when messages arrive before threads are ready to receive. Traditionally, we partition thread-level codes into modelable coarse-grained comp/comm events and perform micro benchmarks on the target architecture to collect training data for surrogate models. We know many sources of uncertainty are introduced due to this coarse-grained abstraction of system detail, but how do we know what is an appropriate level of abstraction? This is a complex question prompting in-depth and ongoing uncertainty quantification (UQ) analysis as our BE methods are refined. In 2015, we laid the groundwork to facilitate this ongoing analysis.

To assist with UQ analysis, we modified the BE simulator discussed in the previous section to enable probabilistic uncertainty propagation via Monte Carlo (MC) simulation. In this mode of execution, uncertain inputs to our BE simulations (i.e., predicted comp/comm event timing from generated surrogate models) are represented as random variables (RV) from an independent and identically distributed probability distribution. The RV distributions are experimentally generated from the same micro benchmark data collected to train our surrogate models. A single pass of the BE simulation is generated by replacing surrogate predictions with samples (with replacement) from the event distributions to produce one possible simulation outcome. This process is repeated many times to produce an output distribution.

Consider the scenario depicted in Figure 8.2. Three threads executing in parallel where thread 0 initializes an array and sends chunks to threads 1 & 2, threads 0-2 perform work on their respective chunks, threads 1-2 communicate modified chunks back to thread 0, then threads 0-2 reduce, finalize, and exit. Repeated execution of real code implementing this scenario on some target system (e.g., 3 nodes of LLNL Cab system) will produce some distribution (e.g., right distribution in Figure 8.3). Traditional BE simulation will treat event timing/ordering as deterministic and produce a single predicted value (left distribution in Figure 8.3). Although the absolute error for this example is relatively low (-1.35%), there is a large discrepancy in the distribution shape (large p-box area in Figure 8.4). Figure 8.5 shows the outcome of a Monte Carlo simulation consisting of 10,000 walks for the same example code. The Monte Carlo simulation is capable of capturing the effect of non-deterministic event ordering on output distributions. Comparing the “true”

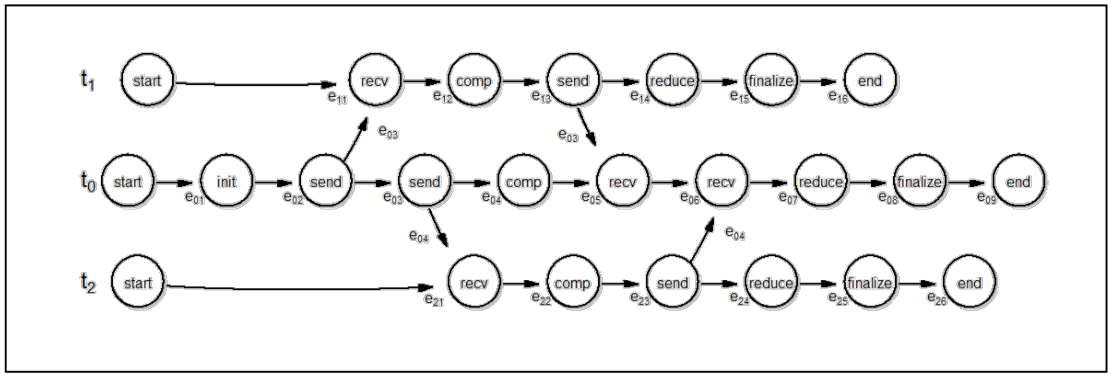


Figure 8.2: Example three thread discrete-event simulation.

distribution from Figure 8.3 with the distribution from Figure 8.5 still shows large discrepancy in the distribution shape, but does show closer agreement than Figure 8.4. In 2016, Further/iterative analysis of BE model characteristics that cause this shape discrepancy will hopefully lead to reduced area shape discrepancy between truth and traditional BE methods.

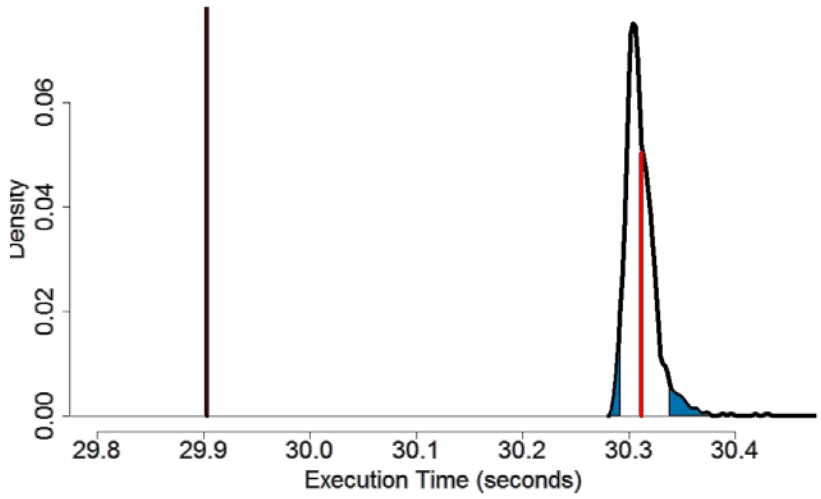


Figure 8.3: Comparison of three thread example from Figure 8.2 targeted to three nodes of LLNL's Cab system. Distribution to the right is a collection of 1,000 random runs while the line to the left is the predicted execution time with our traditional BE methods.

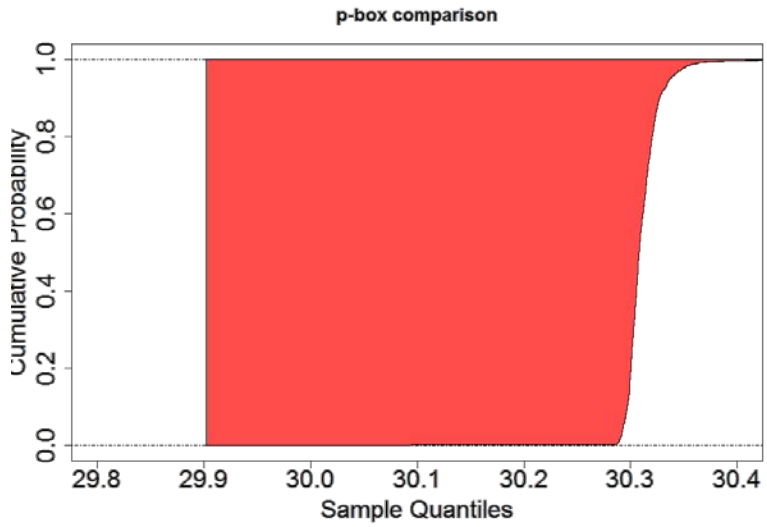


Figure 8.4: p-box comparison between the results from Figure 8.3. Red area represents the discrepancy between 'truth' and prediction with traditional BE methods.

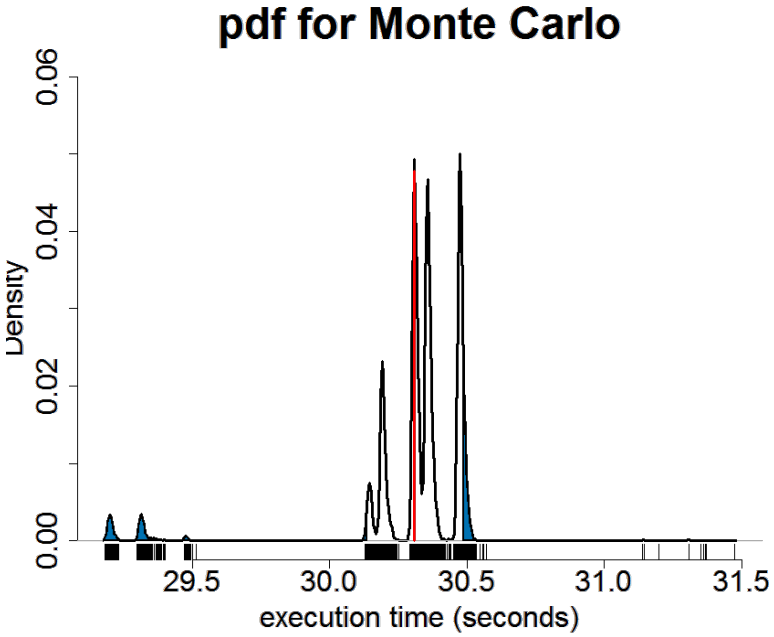


Figure 8.5: Resultant distribution from Monte Carlo simulation of three thread example in Figure 8.2.

8.4 FPGA-based BE Acceleration

It is our belief that it is necessary to go beyond the computing power of conventional parallel software simulator efforts to study the performance and scale of Exascale systems. The goal of this research thrust is to investigate, develop, and optimize an infrastructure to provide FPGA-centric hardware-supported Behavioral Emulation of extreme-scale systems up to and including Exascale. Figure 8.6 shows the landscape of planned FPGA-acceleration studies. In 2014 we worked towards our original project target of a single Exascale simulation distributed over many FPGAs by first focusing on the challenges of efficiently mapping previously discussed BE methods/models onto the University of Florida's Novo-G reconfigurable supercomputer, resulting in an initial single-FPGA prototype (NGEEv1). Although initial testing showed great promise (~100x performance improvement over our BE SMT simulator), the NGEEv1 design was intended as a functional prototype and did not focus on optimization. In 2015 we identified a number issues with the current NGEEv1 design and worked towards design optimization and incorporation of new improvements to the core BE modeling methodology. Issues include:

- appBEO scripts stored as thread-level instruction traces in on-chip block RAMs limiting overall simulation size due to finite BRAM per FPGA. We addressed this issue in the short-term by adding FPGA checkpointing functionality (ability to store/load BEO states from within instruction stream), but continue to research a more long-term solution based on an infinite instruction stream from the host at runtime.
- The original NGEEv1 instruction set architecture (ISA) used for internal processing was very limiting resulting in an unnecessarily long appBEO representation which has become inconsistent with recent developments in BE methodology/functionalities. We worked towards augmenting the existing ISA with a spanning subset of new BE control-flow instructions (already incorporated into our software simulator), which allows for a more succinct representation of thread traces, but negatively impacts our FPGA BEO density; efforts in this area are ongoing as we continue to explore this area-functionality tradeoff.
- Explicit emulation of target communication fabric required that we resynthesize a new bit file whenever studying new network configurations (i.e., whenever new commBEOs are required). We continued to investigate efficient ways to overlay a virtual routing fabric over a fixed fabric that will allow for a single optimized FPGA network to handle all simulated networks. We also began to investigate implications of replacing explicit routing with analytical models.
- app and procBEOs remain idle while waiting for comm instruction processing resulting in less than optimal hardware resource utilization. We began to incorporate the concept of "BEO hyperthreading" by adding the ability for procBEOs to read from multiple instruction streams and switch thread context when stalled for comm events. Ongoing experiments to determine if benefits outweigh costs.

These identified issues and our efforts to address them are represented as "NGEEv1 Enhancements" in Figure 8.6. Although these enhancements demonstrate incremental progress towards our original target, they only explore two dimensions of our potential design space: scale of a single simulation and the total number of FPGAs used (i.e., front most plane in Figure 8.6). In 2015, we extended our design space to consider simultaneous simulations. Extension up to the limit of at most one simulation per FPGA was relatively straight forward by running our single simulation solution in parallel on multiple FPGAs in multiple Novo-G nodes (parameter sweeps) and is potentially useful for CMT-nek design space exploration, but an increase beyond this limit will require a fundamentally new approach.

We propose an approach for accelerating DSE by pipelining simulations. This approach extends our design space along the simultaneous simulation axis in Figure 8.6. Since DSE can be done by executing many configurations of a simulation, the ability to pipeline simulations offers the potential to significantly speedup DSE. This is important to us because accelerated DSE extends our capability for finding optimizations for apps such as CMT-nek on potential Exascale architectures.

The approach is to construct pipelines for our simulations and then vary their configuration as we fill the pipeline. Each configuration (input size, architecture, etc.) allows us to explore different portions of the design space. After some initial latency, the pipeline is filled and will give the output of a differently configured simulation on each cycle. For a large number of simulations, this approach will achieve significant speedup.

To demonstrate a proof-of-concept of the proposed approach, consider an example application doing a simple communication-computation-communication. The data flow graph (DFG) for this application (shown in Figure 8.7) shows the dependencies of the operations. We use this DFG to construct the pipeline in Figure 8.8. The pipeline works by adding the time taken for each operation and communication event to the simulation time while preserving the dependencies in the application. The time taken for each operation is given by the performance models P(send), P(recv), and P(kernel) done in previous work. The configuration of the simulation is used as input by the performance models allowing for simulations of different configurations to be loaded in. Once the pipeline is filled (Figure 8.9), each additional cycle will produce a simulation output for a differently configured simulation, this allows for fast DSE.

There are several foreseeable challenges that will come with taking this approach, which we will focus on in 2016:

- Implicit dependency in our communication models might prevent pipelining. For these cases analytical models will have to be used.
- Max operation of pipeline in Figure 8.8 may not be accurate for all communication; a more advanced operation will need to be developed.

- There are certain pieces of code (i.e. loops with loop-carried dependencies) which can't be pipelined. Models that abstract the behavior will have to be used instead. Fortunately, we will generally model kernels at a level of granularity above individual loop iterations, which should minimize this problem.
- Some simulations may be too large to fit on a single FPGA and resource sharing will need to be implemented in these cases.
- Certain configuration parameters (i.e. #processors) might cause changes in the FPGA circuit resulting in long recompile times. Use of FPGA virtualization research will need to be leveraged for shorter recompile times.

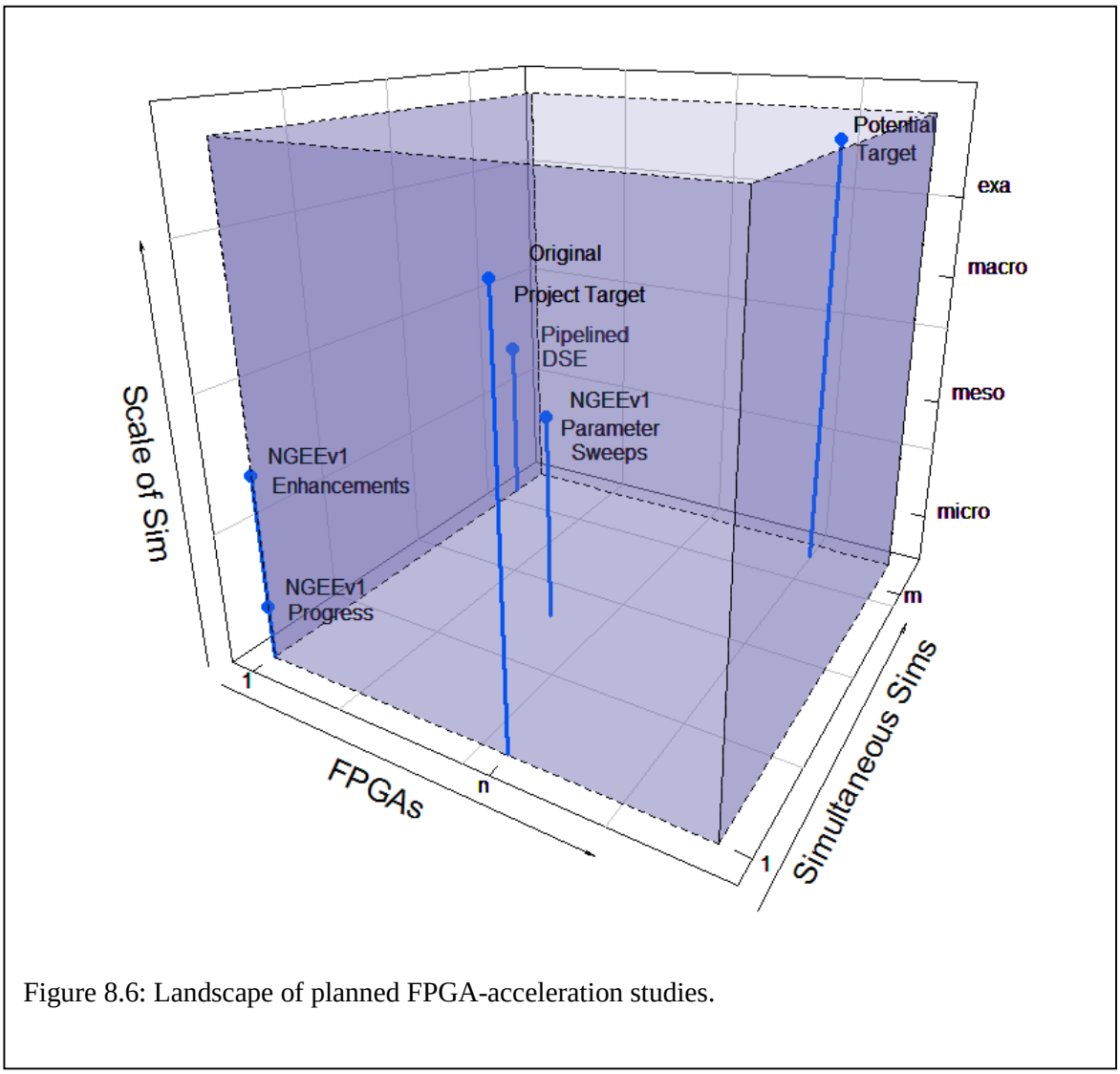


Figure 8.6: Landscape of planned FPGA-acceleration studies.

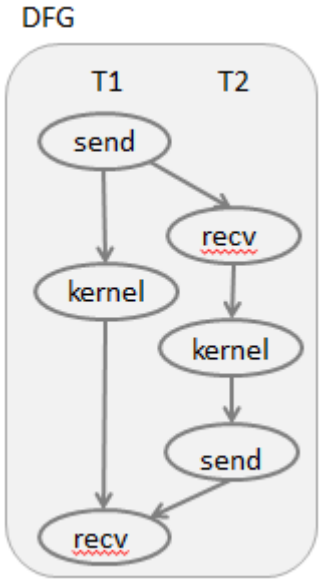


Figure 8.7: DFG for a communication-computation-communication application.

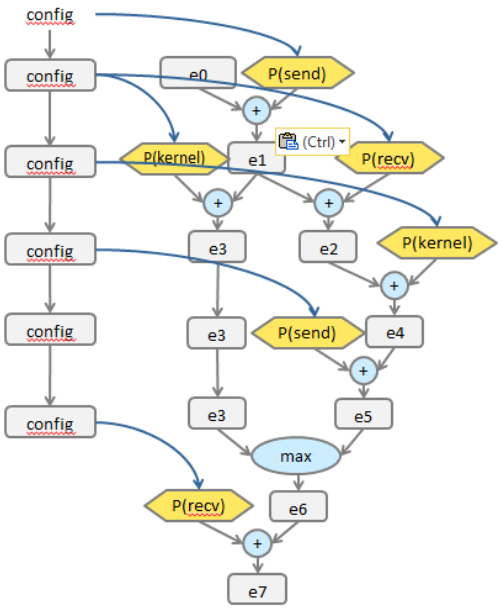


Figure 8.8: Pipeline constructed from DFG in Figure 8.7.

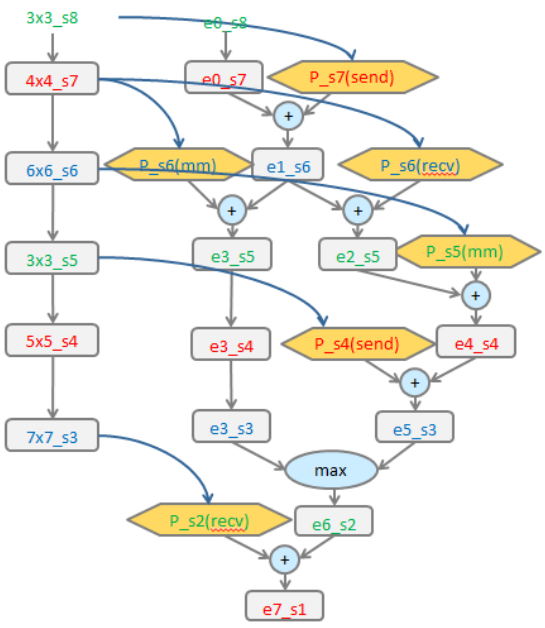


Figure 8.9: Pipeline is filled with different configurations of a simulation and will produce a new output on each cycle.

8.5 Network Performance Characterization

The goal behind this research task is to gain insight via network simulation into the network performance of the code, which can then be used to optimize the CMT-nek software. The focus of this term was to identify an existing simulator that allows the use of abstract application models to drive network simulations, while BE tools are being developed. Lessons learned using this simulator will then be applied to the development of network models in BE simulation platform.

The network characterization research is in its early stages and the results presented here are based on our preliminary exploration of both the application and network models. This work will be extended in 2016 to include a wider set of application and network parameters, network models, and measured performance characteristics.

Simulator selection: The Structural Simulation Toolkit (SST) from Sandia National Laboratory is a PDES framework for simulation of large HPC systems at different levels of granularity. There are two main parts to SST - the Core and the Element libraries. The SST Core is the simulation backend and each Element is a simulator or a model which interfaces with the SST Core.

- We use Ember Element (also, Firefly and Merlin) which contains light-weight end-point models, called Motifs, which can be used to conduct fast simulations. A Motif injects traffic into the network, receives messages, and reacts to them.
- In order to use Ember to simulate an application running on a system, we needed to write a Motif since none existed for CMT-nek. This is a non-trivial task since the application workflow and any communication algorithms are to be condensed into a few lines of code.
- In order to allow simulation of systems with thousands and millions of end-points, we do not simulate processor or memory behavior. Instead, an application Motif contains a time estimate (experimentally determined for a target system) for operations that are local to an end-point.

Developing an abstract application description (CMT-bone):

Since the application code base is massive and is under active development, it is non-trivial to understand and not conducive for fast and extensive performance profiling. To alleviate this issue CMT-nek developers made a first pass at analyzing the key phases in the application and writing a considerably simplified mini-app called CMT-bone. The salient features of the mini-app which we also used as guidelines for our Motif are:

- The three major phases in CMT-nek (hence CMT-bone) involve computing (1) the source terms, (2) the flux divergence, and (3) the numerical flux for all the cubic reference elements.
- The latest version of CMT-nek, and hence CMT-bone, has limited multiphase coupling. As a result, the source terms are set to zero.
- The flux divergence computation is abstracted as the dot product of the gradient operator and the flux vector. It is implemented as the multiplication of the derivative operator matrix (N, N) and the element matrix (N, N, N, Nel) to calculate the partial derivatives along the three

Cartesian coordinate dimensions (r, s, t). Here N is the number of grid points along any one direction in a cubic reference element and Nel is the total number of elements in the computational domain.

- The numerical flux term is computed to enforce boundary conditions necessary to ensure continuity on the element boundaries. The flux divergence is evaluated on the surface of the reference elements which requires surface data exchange between neighboring elements. These exchanges are implemented using a specialized gather-scatter library using either (1) pairwise exchange, (2) crystal-router, or (3) all_reduce onto a big vector.
- The current behavior of CMT-bone can be abstracted into an end-point model or Motif with several iterations of a flux divergence computation phase followed by a nearest-neighbor exchange communication phase. Nearest-neighbor exchange is implemented as simple pairwise exchanges between endpoints, though a crystal-router Motif was developed as well.

Network Simulations: The CMT-bone Motif exposes several application parameters such as element size, physical variables, number of elements per processors, element distribution on each endpoint, and number of threads/endpoints. In our simulations, we varied these application parameters as well as architecture properties such as number of endpoints, packet size, bandwidth, etc. Only a few of these results are shown below in Figure 8.10. The first graph shows the execution time for 1000 timesteps for different element sizes and the second shows the execution time trend with weak scaling (for 1000 timesteps and element size = 10). We are working on adding the capability of simulating more realistic scenarios with load imbalances (within the job) and effect of partitioning schemes of application performance.

Plans for 2016:

In 2016 we plan to continue and extend this work to study other network topologies such as dragonfly and fat-tree networks, effect of routing policies, and different network endpoint capabilities. We also plan to explore the impact of different workload partitioning algorithms on network performance, especially as particles are introduced in the CMT-nek code.

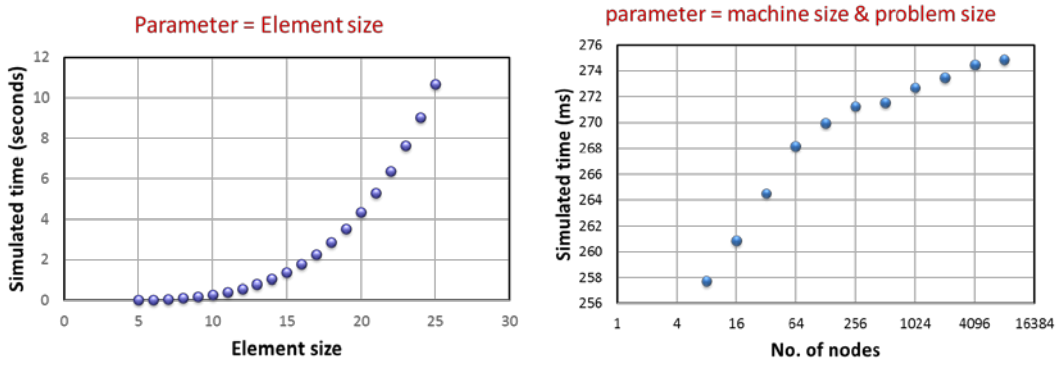


Figure 8.10: Examples of SST simulations for CMT-bone Motif on a 3D torus GigE network.

9. Deep Dive

The University of Florida held a Deep Dive Workshop on Feb 3-4, 2015. The Agenda is presented below, and the presentation slides can be found on the CCMT webpage at:

https://www.eng.ufl.edu/ccmt/events/workshops/

Agenda:

Deep Dive
University of Florida
February 3-4, 2015

Current Attendee List:

Bob Voigt	NNSA HQ	rvoigt@krellinst.org
Matt Bement	LANL	bement@lanl.gov
David Daniel	LANL	ddd@lanl.gov
Dave Nystrom	LANL	wdn@lanl.gov
Maya Gokhale	LLNL	maya@llnl.gov
Martin Schulz	LLNL	schulzm@llnl.gov
Jim Ang	SNL	jaang@sandia.gov
Arun Rodrigues	SNL	afrodri@sandia.gov
Jeremy Wilke	SNL	jjwilke@sandia.gov
S. Balachandar “Bala”	University of Florida	bala1s@ufl.edu
Alan George	University of Florida	george@hcs.ufl.edu
Rafi Haftka	University of Florida	haftka@ufl.edu
Herman Lam	University of Florida	hlam@ufl.edu
Sanjay Ranka	University of Florida	ranka@cise.ufl.edu
Greg Stitt	University of Florida	gstitt@ece.ufl.edu
Tom Jackson	University of Florida	tlj@ufl.edu
Tania Banerjee	University of Florida	tmishra@cise.ufl.edu

University of Florida Students:

Dylan Rudolph	rudolph@hcs.ufl.edu
Nalini Kumar	nkumar@hcs.ufl.edu
Carlo Pascoe	carlo.pascoe@gmail.com
Kasim AlliKasim	kasimalli490@yahoo.com
Chris Hajas	chrishajas@ufl.edu
Mohammed Gadou	mgadou@ufl.edu
Michael Retherford	

UF Deep dive agenda:

Tuesday, February 3, 2015

8:20 Van pickup at Hilton

8:30 – 9:00 Breakfast

9:00 – 9:30 **Welcome and Deep-Dive Overview (3 Sessions)**

1. Behavioral emulation (BE): modeling & simulation/emulation methods
2. CS issues (performance, energy, and thermal)
3. Use of reconfigurable computing to accelerate behavioral emulation

* Each of the three deep-dive sessions is designed to be ***interactive*****:** a combination of short presentations by UF and Tri-lab researchers, intermixed with discussion, demonstrations, etc.

9:30 – 11:30 **Session 1: Behavioral Emulation: Modeling & Simulation/Emulation Methods**

- UF topics:
  - Behavioral characterization
  - Parameter estimation
- Tri-lab topics:
  - Overview of FastForward 2 and DesignForward 2 (Jim Ang, SNL)
  - Multi-scale architectural simulation with the Structural Simulation Toolkit (Arun Rodrigues, SNL)

11:30 – 12:30 Lunch

12:30 – 2:00 **Session 1 (continued): Behavioral Emulation: Beyond Device Level**

- UF topics:
  - Synchronization for speed
  - Congestion modeling
  - Behavioral characterization & modeling beyond device level
- Tri-lab topics:
  - Using discrete event simulation for programming model exploration at extreme-scale (Jeremy Wilke, SNL)
  - ASC next-generation code projects (David Daniel, LANL)

2:00 – 5:00 **Session 2: CS Issues (Performance, Energy, and Thermal)**

- UF topics:
  - Performance and autotuning for hybrid architectures
  - Energy and thermal optimization
  - Dynamic load balancing
- Tri-lab topics:
  - Performance, energy, and thermal benchmarking (Jim Ang, SNL)
  - Why power is a performance issue: utilizing overprovisioned systems (Martin Schulz, LLNL)

* There will be an afternoon coffee break in this time slot

6:30 Dinner (University Hilton)

Wednesday February 4, 2015

8:20 Van pickup

8:30 – 9:00 Breakfast

9:00 – 11:00 **Session 3: Use of Reconfigurable Computing to Accelerate Behavioral Emulation**

- UF topics:
  - Efficient mapping of behavioral emulation objects (BEOs) onto a system of FPGAs
  - Demo of current single FPGA prototype
  - Transitioning to multiple FPGAs
  - Challenges associated with maximizing emulation speed while maintaining scalability/usability
- Tri-lab topic:
  - FPGA-based emulation of processing near memory (Maya Gokhale, LLNL)

11:00 – 12:00 Open discussion and planning for action items

12:00 Box lunch; transportation to airport as needed.

10. Publications

2014

1. Annamalai, S., Neal, C., Ouellet, F., Rollin, B., Jackson, T.L. & Balachandar, S. (2014). Numerical Simulation of Explosive Dispersal of Particles in Cylindrical Geometry. Slides available online on the IWPCTM 2014 website - https://iwpctm.llnl.gov/index.html
2. Kumar, N., Pascoe, C., Rudolph, D., Lam, H., George, A., and Stitt, G. (2014). Multi-scale, Multi-objective, Behavioral Modeling & Emulation of Extreme-scale Systems. Workshop on Modeling & Simulation of Systems & Applications, Seattle, WA, August 13-14, 2014.
3. Zunino, H., Adrian, R.J., Ding, L., and Prestridge, K. (2014). New in-situ, non-intrusive calibration. 67th Annual Meeting of the APS Division of Fluid Dynamics, San Francisco, CA, November 23-25, 2014.
4. Adrian, R.J., Wu, X., Moin, P., Baltzer, J.R. (2014). Osborne Reynolds pipe flow: direct numerical simulation from laminar to fully-developed turbulence. 67th Annual Meeting of the APS Division of Fluid Dynamics, San Francisco, CA, November 23-25, 2014.
5. Adrian, R. co-authored several presentations. These include: *Triple Pulse Particle Image Velocimeter/Accelerometer Measurements of Flow-Structure Interaction* (S. Gogineni), *Effect of Small Roughness Elements on Thermal Statistics of Turbulent Boundary Layer at Moderate Reynolds Number* (A. Doosttalab), *Multi-Scale Coherent Structure Interactions in Rayleigh-Benard Convection* (P. Sakievich), and *New in-situ, non-intrusive calibration* (H.A. Zunino). *Optimization and Application of Surface Segmentation Technique for Tomographic PIV* (L. Ding). 2014 67th Annual Meeting of APS Division of Fluid Dynamics. Bulletin of the American Physical Society, 59.
6. Annamalai, S., Parmar, M., Mehta, Y., & Balachandar, S. (2014). Modeling of hydrodynamic forces on a finite-sized spherical particle due to a planar shock wave. Bulletin of the American Physical Society, 59.
7. Akiki, G., & Balachandar, S. (2014). Immersed Boundary Methods on Non-Uniform Grids for Simulation of a Fully Resolved Bed of Particles in a Near-Wall Turbulent Flow. Bulletin of the American Physical Society, 59.
8. Rollin, B., Annamalai, S., Neal, C., Jackson, T., & Balachandar, S. (2014). Numerical Study of Explosive Dispersal of Particles. Bulletin of the American Physical Society, 59.
9. Jackson, T., Sridharan, P., Zhang, J., & Balachandar, S. (2014). Shock propagation over a deformable particle. Bulletin of the American Physical Society, 59.
10. Matsumura, Y. and Jackson, T.L. (2014). Numerical simulation of fluid flow through random packs of cylinders using immersed boundary method. Physics of Fluids, Vol. 26, 043602.
11. Matsumura, Y. and Jackson, T.L. (2014). Numerical simulation of fluid flow through random packs of polydisperse cylinders. Physics of Fluids, Vol. 26, 123302.
12. Anderson, M.J., Jackson, T.L., Wasistho, B., and Buckmaster, J. (2014). A physics-based hot-spot model for pore collapse in HMX. 46th JANNAF Combustion Subcommittee Meeting, Albuquerque, NM, December 8-11, 2014.
13. Annamalai, S., Parmar, M. K., Ling, Y., & Balachandar, S. (2014). Nonlinear Rayleigh–Taylor Instability of a Cylindrical Interface in Explosion Flows. Journal of Fluids Engineering, 136(6), 060910.

14. Annamalai, S., Balachandar, S., & Parmar, M. K. (2014). Mean force on a finite-sized spherical particle due to an acoustic field in a viscous compressible medium. Physical Review E, 89(5), 053008.
15. Mankbadi, M. R., & Balachandar, S. (2014). Multiphase effects on spherical Rayleigh-Taylor interfacial instability. Physics of Fluids (1994-present), 26(2), 023301.
16. Chen Q., R. J. Adrian, Q. Zhong, D. Li, X. Wang (2014). "Experimental study on the role of spanwise vorticity and vortex filaments in the outer region of open-channel flow", J. Hydraulic Res., 1-14.
17. Hengxing Tan and Sanjay Ranka (2014). Thermal-aware Scheduling for Data Parallel Workloads on Multi-Core Processors, Proceedings of ISCC 2014.

2015

18. Matsumura, Y., Jenne, D. and Jackson, T.L. (2015). Numerical simulation of fluid flow through random packs of ellipses. Physics of Fluids, Vol. 27, 023301.
19. Sridharan, P., Jackson, T.L., Zhang, J. and Balachandar, S. (2015). Shock interaction with 1-D array of particles in air. Journal of Applied Physics, Vol. 117, 075902.
20. Jackson, T.L., Buckmaster, J., Zhang, J. and Anderson, M. (2015). Pore collapse in an energetic material from the micro-scale to the macro-scale. Combustion Theory and Modeling, Vol. 19(3), pp. 347-381.
21. Thakur, S., Neal, C., Mehta, Y., Sridharan, P., Jackson, T.L. and Balachandar, S. (2015). Microscale Simulations of Shock Interaction with Large Assembly of Particles for Developing Point-Particle Models. SHOCK15 Meeting, American Physical Society.
22. Gillman, A., Amadio, G., Matous, K. and Jackson, T.L. (2015). Third-order thermos-mechanical properties for packs of Platonic solids using statistical micromechanics. Proceedings of the Royal Society of London A, Vol. 471, 20150060.
23. Annamalai, S. and Balachandar, S. (2015). Generalized Faxen's theorem in time-domain for unsteady viscous compressible flows over a sphere, to be submitted to Phys. Rev. Lett.
24. Schwarzkopf, J.D., Balachandar, S. and Buttler, W.T. (2015). Compressible Multiphase Flows, in Handbook of Multiphase Flows, Ed. Stathis Michalides, CRC Press (to appear).
25. McGrath, T.P., St Clair, J.G. and Balachandar, S. (2015). A compressible multiphase model for dispersed particle flows with application from dense to dilute regimes, submitted Phys. Fluids.
26. Annamalai, S. and Balachandar, S. (2015). Acoustic levitation force on a finite-sized rigid particle, droplet or bubble in a viscous compressible medium. Submitted to Phys. Fluids.
27. Cook, C.R., Balachandar, S. and Chung, J.N. (2015). A modified Characteristic-Based Split (CBS) projection method for compressible flows of real fluids, to be submitted to J. Comput. Phys.
28. Sakievich, P.J., Peet Y.T., Adrian, R.J. (2015). Large-scale, coherent structures in wide-aspect ratio, turbulent, Rayleigh-Benard convection. Ninth International Symposium on Turbulence and Shear Flow Phenomena. July 2015.
29. Wu, X., Moin. P, Adrian, R.J., Baltzer, J.R. (2015). Osborne Reynolds pipe flow: Direct simulation from laminar through gradual transition to fully developed turbulence. Proc Natl Acad Sci U.S.A. 2015 Jun 30;112(26):7920-4. DOI: 10.1073/pnas.1509451112. Epub 2015 Jun 15.

30. Chojnicki K. N., A. B. Clarke, J.C. Phillips, R. J. Adrian (2015). The evolution of volcanic plume morphologies from short-lived eruptions. Geology, v. 43 no. 8 p. 707-710. July 10, 2015, DOI:10.1130/G36642.1.
31. G. Akiki, T Jackson, S Balachandar (2015). Mean and Fluctuating Force Distribution in a Random Array of Spheres. Bulletin of the American Physical Society, Vol. 60.
32. TL Jackson, P Sridharan, J Zhang, S Balachandar (2015). Numerical Simulation of Shock Interaction with Deformable Particles Using a Constrained Interface Reinitialization Scheme. Bulletin of the American Physical Society, Vol. 60.
33. J. Zhang, T.L. Jackson (2015). Detonation initiation with thermal deposition due to pore collapse in energetic materials - Towards the coupling between micro- and macroscales. Bulletin of the American Physical Society, Vol. 60.
34. Zhang, J. and Jackson, T.L. (2015). Detonation Initiation with Thermal Deposition due to Pore Collapse in Energetic Materials - Towards the Coupling between Micro- and Macroscale. AIAA Paper No. 2015-4097, 51st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 27-29 July 2015, Orlando, FL.
35. Amadio, G. and Jackson, T.L. (2015). A new packing code for creating microstructures of propellants and explosives. AIAA Paper No. 2015-4098, 51st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 27-29 July 2015, Orlando, FL.
36. Diggs, A., Balachandar, S. (2015). Evaluation of Methods for calculating volume fraction in Eulerian-Lagrangian multiphase flow simulations, submitted to J. Comput. Phys.
37. Diggs, A., Balachandar, S. (2015). Modeling and Simulation Challenges in Eulerian-Lagrangian Computations of Shock-Driven Multiphase Flows, Bulletin of the American Physical Society, Vol. 60.
38. Park, C., Haftka, R.T. and Kim, N. (2015). The Effect of Ignoring Dependence between Failure Modes on Evaluating System Reliability. Submitted to Structural Multi-disciplinary Optimization.
39. Park, C., Choi, J., and Haftka, R.T. (2015). Teaching a Verification and Validation Course using Simulations and Experiments with Paper Helicopter, submitted to The Journal of Verification, Validation and Uncertainty Quantification (2015).
40. Chanyoung Park, R. T. Haftka and Nam H. Kim, (2015). “Remarks on Multi-fidelity Surrogates” Submitted to Structural and Multidisciplinary Optimization.
41. Chanyoung Park and Nam H. Kim, (2015). “Safety Envelope for Load Tolerance of Structural Element Design based on Multi-stage Testing” Submitted to Advances in Mechanical Engineering.
42. Banerjee, T., Ranka, S., (2015). A Genetic Algorithm Based Autotuning Approach for Performance and Energy Optimization, The 6th International Green and Sustainable Computing Conference, 2015.
43. Banerjee, T., Gadou, M., Ranka, S., (2015). A Genetic Algorithm Based Approach for Multi-objective Hardware/Software Co-optimization, submitted to Sustainable Computing, Informatics and Systems.
44. Banerjee, T., Rabb, J., Ranka, S., (2015). Performance and Energy Benchmarking of Spectral Solvers on Hybrid Multicore Machines, submitted to Sustainable Computing, Informatics and Systems.

45. Banerjee, T. and Ranka, S. (2015). Power modeling and predicting method on multi-core processors. In preparation.
46. N. Kumar, M. Shringarpure, T. Banerjee, J. Hackl, S. Balachandar, H. Lam, A. George, S. Ranka, (2015). CMT-bone: A mini-app for Compressible Multiphase Turbulence Simulation Software. Workshop on Representative Applications, co-located with IEEE Cluster 2015, Chicago, IL, USA, Sep 8-11 2015.
47. Dylan Rudolph and Greg Stitt (2015). An Interpolation-Based Approach to Multi-Parameter Performance Modeling for Heterogeneous Systems, IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), July 2015.

2016

48. Annamalai, S., Rollin, B., Ouellet, F., Neal, C., Jackson, T.L. and Balachandar, S. (2016). Effects of initial perturbations in the early moments of an explosive dispersal of particles. Proceedings of the IWPCTM 2014, ASME Journal of Fluids Engineering, accepted.
49. F. Ouellet, S. Annamalai, and B. Rollin. (2016). "Effect of a Bimodal Initial Particle Volume Fraction Perturbation in an Explosive Dispersal of Particles", *Proceedings of the 19th Biennial APS Conference on Shock Compression of Condensed Matter*, accepted.
50. Doosttalab, A., Araya, G., Newman, J., Adrian, R.J., Jansen, K., Castillo, L. (2016). Effect of small roughness elements on thermal statistics of a turbulent boundary layer at moderate Reynolds number. J Fluid Mech. vol 787, pp. 84-115. DOI: 10.1017/jfm.2015.676.
51. Sridharan, P., Jackson, T.L., Zhang, J., Balachandar, S., and S. Thakur (2016). Shock interaction with deformable particles using a constrained interface reinitialization scheme. Journal of Applied Physics, Vol. 119, 064904, 18 pages.
52. Mehta, Y., Jackson, T.L., Zhang, J. and Balachandar, S. (2016). Numerical investigation of shock interaction with one-dimensional transverse array of particles in air. Journal of Applied Physics, submitted.
53. Akiki, G., and Balachandar, S. (2016). Immersed boundary method with non-uniform distribution of Lagrangian markers for a non-uniform Eulerian mesh. Journal of Computational Physics. Vol. 307, pp. 34-59. DOI 10.1016/j.jcp.2015.11.019.

11. Conferences and Presentations

2014

1. S. Balachandar (2014). "Center for Compressible Multiphase Turbulence – Overview", PSAAP-II Kickoff Meeting, Albuquerque, NM, December 9-11, 2013.
2. Sanjay Ranka, Herman Lam (2014). CCMT – Extreme Scale CS Research, PSAAP-II Kickoff Meeting, Albuquerque, NM, December 9-11, 2013.
3. S. Balachandar (2014). "Center for Compressible Multiphase Turbulence – Overview", ASC PI Meeting, Kansas City, MO, February 26, 2014.
4. S. Balachandar (2014). "Center for Compressible Multiphase Turbulence – Overview and CMT Physics", CCMT Road trip to Sandia National Laboratories, Albuquerque, NM, March 24, 2014.

5. R.T. Haftka (2014). "Center for Compressible Multiphase Turbulence – V&V and Uncertainty Quantification Research and Innovation", CCMT Road trip to Sandia National Laboratories, Albuquerque, NM, March 24, 2014.
6. Herman Lam (2014). "Center for Compressible Multiphase Turbulence – Exascale Emulation", CCMT Road trip to Sandia National Laboratories, Albuquerque, NM, March 24, 2014.
7. S. Balachandar (2014). "Center for Compressible Multiphase Turbulence – Overview and CMT Physics", CCMT Road trip to Los Alamos National Laboratory, Los Alamos, NM, March 25, 2014.
8. R.T. Haftka (2014). "Center for Compressible Multiphase Turbulence – V&V and Uncertainty Quantification Research and Innovation", CCMT Road trip to Los Alamos National Laboratory, Los Alamos, NM, March 25, 2014.
9. Herman Lam (2014). "Center for Compressible Multiphase Turbulence – Exascale Emulation", CCMT Road trip to Los Alamos National Laboratory, Los Alamos, NM, March 25, 2014.
10. S. Balachandar (2014). "Fundamental Advances in Compressible Multiphase Flows – Potential Relevance to Liquid Atomization and Spray Systems", Keynote Lecture, Institute for Liquid Atomization and Spray Systems, Portland, OR, May 18-21, 2014.
11. S. Balachandar (2014). "The Need for Fundamental Advances in Compressible Multiphase Flows – Shock-particle Interaction to Explosive Dispersal", Department of Aerospace Engineering, Iowa State University, Ames, Iowa, March 6, 2014
12. S. Balachandar (2014). "Shock-Particle Interaction to Explosive Particle Dispersal – What Fundamental Advances in Compressible Multiphase Flows are Needed", University of Stuttgart, Stuttgart, Germany, May 7, 2014.
13. Subramanian Annamalai (2014). University of Florida, "Rocflu – An Overview", CCMT Seminar, 2:00 P.M., Wednesday, February 5, 2014, 210 MAE-B.
14. Jackson, T.L. (2014). 67th Annual Meeting, Division of Fluid Dynamics, American Physical Society, San Francisco, CA, Nov 2014. ``Shock propagation over a deformable particle".
15. S. Annamalai, C. Neal, F. Ouellet, B. Rollin, T.J. Jackson, & S. Balachandar (2014). Numerical Simulation of Explosive Dispersal of Particles in Cylindrical Geometry, IWPCTM 2014, San Francisco, California, USA.
16. Heather Zunino (2014). H. Zunino, R.J. Adrian, L. Ding, K. Prestridge, "New in-situ, non-intrusive calibration", 67th Annual Meeting of the APS Division of Fluid Dynamics, San Francisco, CA, November 23-25, 2014.
17. Ronald Adrian: R.J. Adrian, X. Wu , P. Moin , J.R. Baltzer. (2014). Osborne Reynolds pipe flow: direct numerical simulation from laminar to fully-developed turbulence, 67th Annual Meeting of the APS Division of Fluid Dynamics, San Francisco, CA, November 23-25, 2014.
18. B. Rollin, S. Annamalai, C. Neal, T.J. Jackson, & S. Balachandar (2014). Numerical Study of Explosive Dispersal of Particles*;* DFD2014, Bulletin of the American Physical Society, Vol. 59, Number 20 (November 2014).
19. Chanyoung Park, Joo-Ho Choi, Raphael T. Haftka (2014). Teaching a Verification and Validation Course using Simulations and Experiments with Paper Helicopter, *ASME 2014 Verification and Validation Symposium, Las Vegas, NV, May 7-9, 2014*

20. Chanyoung Park (2014). University of Florida, "Dakota – An Overview", CCMT Seminar, 3:00 P.M., Tuesday, April 8, 2014, 221 MAE-A.
21. Behavioral Modeling & Emulation of Extreme-scale Systems, Workshop on Modeling & Simulation of Systems & Applications, Seattle, WA, August 13-14, 2014.
22. Invited Talk. Prof. Nam-Ho Kim (2014). Prioritizing Efforts to Reduce Uncertainty in System Models based on Uncertainty Budget, *Seoul National University, Seoul, Korea, July 7, 2014.*

2015

23. A.M.D. Jost_ and J. Zhang (2015). Numerical Study of Intermittent Laminar Bubble Bursting and Vortex Shedding on an NACA 643-618 Airfoil, AIAA Meeting, 2015.
24. B. Rollin, S. Annamalai, F. Ouellet (2015). A Study of Interfacial Instability in Explosive Dispersal of Particles, 19th Biennial APS Conference on Shock Compression of Condensed Matter, Tampa, FL (2015).
25. B. Rollin, F. Ouellet, S. Annamalai, & S. Balachandar (2015). Numerical Study of Explosive Dispersal of Particles, DFD2015, Bulletin of the American Physical Society, Vol. 60, Number 21 (November 2015).
26. B. Rollin (2015). *Center for Compressible Multiphase Turbulence,* Stewardship Science Academic Programs (SSAP) Symposium, Santa Fe, NM.
27. B. Rollin (2015). Progress on Simulation of Explosive Dispersal of Particles, LANL, Los Alamos, NM.
28. B. Rollin (2015). Toward Predictive Complex Flow Simulations, invited seminar, Embry-Riddle Aeronautical University, Daytona Beach, FL.
29. Zunino, H., Adrian, R.J., Clarke, A.B. (2015). Experimental studies of gas-particle mixtures under sudden expansion. 68th Annual Meeting of the APS Division of Fluid Dynamics, Boston, MA, November 22-24, 2015.
30. Zunino, H., Adrian, R.J., Ding, L. (2015). Non-intrusive calibration technique. Particle Image Velocimetry 2015 meeting, Santa Barbara, CA, September 14-16, 2015.
31. Chanyoung Park, R. T. Haftka, and N. H. Kim (2015). Experience with Several Multi-fidelity Surrogate Frameworks, 11th World Congresses of Structural and Multidisciplinary Optimization, Sydney, Australia, Jun, 2015.
32. Yiming Zhang, N. H. Kim, C. Park, R. T. Haftka (2015). One-dimensional Function Extrapolation Using Surrogates, 11th World Congresses of Structural and Multidisciplinary Optimization, Sydney, Australia, Jun, 2015.
33. Yiming Zhang, N. H. Kim, R. T. Haftka and C. Park (2015). Function Extrapolation at One Inaccessible Point Using Converging Lines, ASME 2015 International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Boston, MA, USA, August 2-5, 2015.
34. Banerjee, T. (2015). A Genetic Algorithm Based Autotuning Approach for Performance and Energy Optimization, 6th International Green and Sustainable Computing Conference, Las Vegas, NV, Dec 2015.
35. Banerjee, T. (2015). Optimizing Nek5000 kernels for performance and energy, on May 27, 2015 at LLNL.

36. N. Kumar, A. George, H. Lam, G. Stitt, S. Hammond (2015). Understanding Performance and Reliability Trade-offs for Extreme-scale Systems using Behavioral Emulation, 2015 Workshop on Modeling & Simulation of Systems and Applications (ModSim 2015), Seattle, Washington.
37. N. Kumar, C. Hajas, A. George, H. Lam, G. Stitt (2015). Multi-scale, Multi-objective Behavioral Emulation of Future-gen Applications and Systems, 2015 Salishan Conference on High-Speed Computing, Gleneden Beach, Oregon, April 27-30, 2015.
38. "Scalable Network Simulations", Random access talk, 2015 Salishan Conference on High-Speed Computing, Gleneden Beach, Oregon.
39. Zhang, J. and Jackson, T.L. (2015). Detonation Initiation with Thermal Deposition due to Pore Collapse in Energetic Materials - Towards the Coupling between Micro- and Macroscale. 51st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 27-29 July 2015, Orlando, FL.
40. Amadio, G. and Jackson, T.L. (2015). A new packing code for creating microstructures of propellants and explosives. 51st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 27-29 July 2015, Orlando, FL.
41. G. Akiki, T Jackson, S Balachandar (2015). Mean and Fluctuating Force Distribution in a Random Array of Spheres. 68th Annual Meeting, Division of Fluid Dynamics, APS, Boston, MA, Nov. 2015.
42. TL Jackson, P Sridharan, J Zhang, S Balachandar (2015). Numerical Simulation of Shock Interaction with Deformable Particles Using a Constrained Interface Reinitialization Scheme. 68th Annual Meeting, Division of Fluid Dynamics, APS, Boston, MA, Nov. 2015.
43. J. Zhang, T.L. Jackson (2015). Detonation initiation with thermal deposition due to pore collapse in energetic materials - Towards the coupling between micro- and macroscales. 68th Annual Meeting, Division of Fluid Dynamics, APS, Boston, MA, Nov. 2015.
44. Dylan Rudolph and Greg Stitt (2015). An Interpolation-Based Approach to Multi-Parameter Performance Modeling for Heterogeneous Systems, IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), July 2015.

2016

45. Yiming Zhang, N. H. Kim, Chanyoung Park, and R. T. Haftka (2016). Function Extrapolation of Noisy Data using Converging Lines, AIAA Modeling and Simulation Technologies Conference, San Diego, CA, USA, 4-8 January 2016.
46. Chanyoung Park, Giselle Fernández-Godino, R. T. Haftka, and N. H. Kim (2016). Validation, Uncertainty Quantification and Uncertainty Reduction for a Shock Tube Simulation, 18th AIAA Non-Deterministic Approaches Conference, San Diego, CA, USA, 4-8 January 2016.
47. Giselle Fernández-Godino, Angela Diggs, Chanyoung Park, N. H. Kim, and R. T. Haftka (2016). Anomaly Detection Using Groups of Simulations, 18th AIAA Non-Deterministic Approaches Conference, San Diego, CA, USA, 4-8 January 2016.

12. Workshops Held or Attended

2014

1. Dr. Chanyoung Park. Attended "A Short Course on Uncertainty Quantification", Stanford CA, June 2-4 2014.
2. Tania Banerjee. Salishan Conference, April 2014.
3. 2014 International Workshops on the Physics of Compressible Turbulent Mixing, hosted by Lawrence Livermore National Laboratory, 31 August 2014 - 5 September 2014, San Francisco, California, USA.
4. 67th Annual Meeting of the APS Division of Fluid Dynamics, November 23–25, 2014 - San Francisco, California.
5. Professor Nam-Ho Kim. Attended, "A Short Course on Uncertainty Quantification", Stanford CA, June 2-4 2014.
6. Heather Zunino presented tomographic PIV at a PIV workshop to a group of faculty, post-docs, and Ph.D. students at Instituto Superior Técnico (IST) in Lisbon, Portugal. She was invited to stay for this workshop and present after meeting with several researchers from IST at the Laser Symposium this summer.

2015

7. Presented "Dakota - Tutorial", CCMT workshop, 4:00 P.M., Thursday, February 19, 2015, Feb 19 2015.
8. S. Balachandar and T.L. Jackson. Co-organized a mini-symposium titled, Turbulence and Mixing in Shock-Driven Multiphase Flows. APS (APS) Topical Group on Shock Compression of Condensed Matter (SCCM), Tampa, Florida, June 2015.
9. Deep Dive Workshop. Held at the University of Florida on Feb 3-4, 2015.
10. "Good Software Engineering Practices and Beyond" Workshop - Internal workshop - organized by Bertrand Rollin - Macroscale team, held Feb 19, 2015.
11. Heather Zunino presented tomographic PIV at a PIV workshop to a group of faculty, post-docs, and Ph.D. students at Instituto Superior Técnico (IST) in Lisbon, Portugal. She was invited to stay for this workshop and present after meeting with several researchers from IST at the Laser Symposium summer 2015.
12. Tania Banerjee Attended the 6th International Green and Sustainable Computing Conference, 2015.

2016

13. Dr. Sanjay Kumar and Mr. Carlo Pascoe (2016). Attended the WEST workshop. Dr. Kumar gave a talk.

13. Students and Staff Internships

13.1 Internships Completed

2014

1. Heather Zunino, Ph.D. Student (ASU), US, Dr. R. Adrian. Ms. Zunino completed her 10-consecutive week stay at Los Alamos National Laboratory, under the mentorship of Dr. Kathy Prestridge this summer (May-August 2014). Dr. Prestridge is the Extreme Fluids Team leader in the Physics-23 group at LANSCE. Project: Vertical Shock Tube (Calibration and Tomographic PIV), Horizontal Shock Tube (Particle Tracking Program).
2. Kevin Cheng, MS Student, Florida. Lawrence Livermore National Laboratory. Mentor: Dr. Maya Gokhale, Dr. Scott Lloyd. Project: An Emulation Framework for Tracing near Memory Computation. US, Dr. Alan George, ECE, MS (graduated Fall 2014), core.
3. Dr. Chanyoung Park, Postdoc, CCMT. Visited Sandia National Laboratories, Albuquerque NM, on March 24-28 2014.

2015

4. Dr. Jason Hackl, Postdoc, CCMT. Visited LLNL from February 23-27, 2015. LLNL. Sam Schofield, Robert Nourgaliev, Rob Rieben, Tzanio Kolev, Fady Najjar, David Dawson. CMT-nek.
5. Dr. Bertrand Rollin, Staff Scientist, CCMT. March 16-20, 2015, LANL.
6. Nalini Kumar, Ph.D. Student, India, ECE, Dr. Alan George, part cost share and part leveraged. (Internship not required). March-May, 2015. LLNL. Dr. James Ang.
7. Chris Hajas, M.S. Student, US, ECE, Dr. Herman Lam, core. May 18-August 18, 2015 at LLNL with Dr. Maya Gokhale.
8. Christopher Neal, Ph.D. Student, US, MAE, Dr. S. Balachandar, core. June 14-August 20, 2015 at LLNL with Dr. Kambiz Salari. [Chris Neal (graduate student) spent 10 weeks at Lawrence Livermore National Lab (LLNL) over summer 2015. He was paired with Dr. Kambiz Salari as his mentor who had similar research interests. The work that was performed will not be described here. He was asked by Dr. Salari to return to LLNL next summer for a second internship at the lab. Chris' work will also be highlighted in the Stewardship Science Academic Programs (SSAP) Annual, which highlights the exceptional research activities and people supported by the SSAP (PSAAP is under the SSAP).]
9. Carlo Pascoe, Ph.D. Student, US, ECE, Dr. Herman Lam, core. Will intern summer 2015 at LLNL with Dr. Maya Gokhale.
10. Giselle Fernandez, Ph.D. Student, Argentina, MAE, Drs. Haftka and Kim, core. Visited Sandia National Laboratories, Albuquerque NM, Oct 12-Dec 25, 2015.
11. Tania Banerjee, PhD., one week internship in May, 2015 to LLNL.

13.2 Internships Planned

1. Dr. Chanyoung Park, Postdoc, CCMT. Feb 22-26, 2016 at LLNL with Dr. Samuel P. Schofield.

2. Paul Crittenden, Ph.D. Student, US, MAE, Dr. Balachandar, core. Internship planned to LANL for summer 2016 with Dr. Marianne Francois.
3. Georges Akiki, Ph.D. Student, US, MAE, Dr. Balachandar, core. Internship planned to LANL for summer 2016 with Dr. Marianne Francois.
4. David Zwick, Ph.D., US, MAE, Dr. Balachandar, core. Internship planned to Sandia National Lab for summer 2016.

13.3 Internships Not Yet Planned

1. Fred Ouellet, Ph.D. Student, US, MAE, Dr. S. Balachandar, core
2. Prashanth Sridharan, Ph.D., US, MAE, Dr. Balachandar, core
3. Dr. Tania Banerjee, Center Staff, US, core (one week required)
4. Goran Marjanovic, Ph.D., MAE, Dr. Balachandar, core
5. Yash Metha, Ph.D., MAE, Dr. Balachandar, cost share (internship not required)
6. Yiming Zhang, Ph.D., China, MAE, Drs. Haftka and Kim, cost share (internship not required)
7. Kyle T. Hughes, Ph.D. Student, US, MAE, Drs. Haftka and Kim, core
8. Sam Nili, Ph.D. Student, US, MAE, Drs. Haftka and Kim, core
9. Brad Durant, Ph.D., MAE, Dr. Balachandar, core
10. Kasim Alli, Ph.D. Student, US, ECE, Dr. Greg Stitt, core
11. Ryan Blanchard, BS, US, ECE, Dr. Herman Lam, core
12. Dylan Rudolph, Ph.D. Student, US, ECE, Dr. Greg Stitt, part core and part leveraged
13. Ting-Hsin Kuo: Electrical and Computer Engineering, UF, core
14. Mohamed Gadou, Ph.D., Dr. Sanjay Ranka, Computer Science, UF, core

13.4 Graduated Students

1. Kevin Cheng, MS Student (2014). Dr. Alan George, ECE.
2. Hugh Miles, BS Student (2015). US, ECE, Dr. Greg Stitt.
3. Chris Hajas, M.S. Student (2015). US, ECE, Dr. Herman Lam.
4. Angela Diggs, Ph.D. (2015). US, MAE, Dr. S. Balachandar (other funding, internship not required). Currently employed at Eglin AFB and working with the Center.
5. Subramanian Annamalai, Ph.D. (2015). MAE, Dr. S. Balachandar. Currently postdoc in the Center.

13.5 Placement of Staff

1. Dr. Bertrand Rollin, Staff Scientist, CCMT. Faculty position at Embry-Riddle, Fall 2015.
2. Dr. Mrugesh Shringarpure, Postdoc, CCMT. Researcher, Exxon Mobil, Spring 2016.

14. NNSA Laboratory Interactions

2014

1. Rob Cunningham, LANL. Setting up "Florida" file sharing group on Mustang
2. Blaise Barney, LLNL. Setting up account on Vulcan and Surface.
3. Greg Weir, SNL. Introduction to Catalyst.
4. Nathan Fabian, SNL. Introduction to Catalyst.
5. Don Frederick, LLNL. Issue with submitting a run on Vulcan.
6. John Gyllenhaal, LLNL. Help building and running Rocflu on Vulcan.
7. Jan Nunes, LLNL. Account request on Edge
8. Discussions with Donald Frederick of Lawrence Livermore National Laboratory related to MPI issues on the Vulcan computing cluster – discussion was relevant to parallel operation of the Rocflu code.
9. Telecon with Paraview Catalyst developers and users (Greg Weirs, Nathan Fabian, Kenneth Moreland at Sandia National Laboratory) at about deploying the Catalyst library into Rocflu for in-situ visualization.
10. Worked with Greg Lee (Livermore Computing Center)–who is a software debugging/ troubleshooting expert at LLNL–to get Rocflu to run at scale on the Vulcan computing cluster at LLNL.
11. Interactions with Rich Cook (Livermore Computing) who is the visualization expert at LLNL–we are working with him to get in-situ visualization capabilities using Catalyst integrated into Rocflu.
12. Interacted with David DeBonis at Sandia National Laboratories to get setup with using PowerInsight for power and energy measurements.
13. Interacted with Robert Cunningham and Amanda Bonnie at Los Alamos National Laboratory for temperature measurements using the data collection tool Lightweight Distributed Metric Service.
14. Interacted with Justin A. Too and Daniel J. Quinlan at the Lawrence Livermore National Laboratory on compilation and installation of ROSE.
15. Interacted with Blaise Barney at the Lawrence Livermore National Laboratory on various system issues, including access to CVS.
16. Road trip to SNL (Albuquerque) and LANL, March 24 – 26, 2014 (Herman, Bala, and Rafi)
17. SNL (Albuquerque: Jim Ang, Arun Rodrigues, Scott Hemmert, Simon Hammond - from SST team at SNL, Albuquerque, NM
18. SNL (Livermore): Jeremiah Wilke from SST Macro team at SNL, Livermore, CA

2015

19. LLNL: Maya Gokhale regarding proposal to analyze memory access demands of CMT-Nek kernels and evaluate the potential benefits of utilizing stacked memories with embedded logic functions to increase performance
20. Dr. Steve Beresh (SNL), visit CCMT and gave talk, Thursday April 23, 2015
21. The members of the microscale group (in particular Chris Neal and Yash Mehta, another graduate student) have worked closely with the staff at LLNL's Livermore Computing center to resolve an issue that Rocflu was having when it was scaled up and executed on

the BG/Q machine Vulcan. We have also been in touch with Rich Cook at LLNL's Livermore Computing to continue our work with Paraview's Catalyst library integration into Rocflu. During the fall and spring, Chris Neal has been in touch with Dr. Kambiz Salari to keep him up-to-date on the progress of Chris's research. Chris and Yash were aided by Blaise Barney (LLNL) in renewing their computing accounts at LLNL.

22. Drs. Jason Hackl and Mrugesh Shringarpure and David Zwick and Goran Marjanovic. Visited Argonne National Lab for the 2015 Nek User/Developer meeting to commit CMT-nek to the nek5000 repository and work intensively with Prof. Paul Fischer.
23. Barry Rountree, LLNL. Tania worked closely with Barry's group to come up to speed on using RAPL to measure and bound power on Intel platform
24. Tanzima Islam, LLNL. Tania is working with Tanzima on validating CMT-bone proxy app
25. David DeBonis, SNL. Tania interacted with David to come up to speed on using PowerInsight for power and energy measurements.
26. Patrick Lavin, a summer intern with Barry Rountree, worked on the performance issue and used vectorization effectively on the derivative optimization code getting about 40% improvement using vectorization alone. Tania will follow up with verifying if it is possible to get further improvement with CHiLL based autotuning approach.
27. S. Balachandar and H. Lam attended and presented at the 2015 ASC PI meeting at Monterey, CA.
28. S. Balachandar visited LANL and interacted with Drs. Kathy Prestridge, Robert Gore, and John Schwarkopf in November 2015.

2016

29. S. Balachandar and S. Ranka attended and presented at the 2016 ASC PI meeting at Las Vegas, NV.