Quantifying Uncertainties in Radiation Hydrodynamics Models

Krzysztof Fidkowski
Department of Aerospace Engineering
University of Michigan
1320 Beal Avenue, 3029 FXB
Ann Arbor, MI 48109
USA

kfid@umich.edu

CRASH PIs: R.P. Drake, M.L. Adams, J.P. Holloway, K.G. Powell, Q. Stout

ABSTRACT

The purpose of the Center for Radiative Shock Hydrodynamics was to test methods that evaluate predictive capabilities for modeling complex phenomena. The phenomenon of interest was a radiation hydrodynamics experiment and the predictor was a multiphysics computer code. CRASH completed its primary objectives in the experimental, numerical simulation, and predictive science facets of the project. These included an assessment of the ability to predict the behavior of a complex system based on simulation data and measurements of simpler systems. This manuscript summarizes key aspects of the project, focusing on the uncertainty quantification, calibration, and solution verification methods that formed the foundation of the predictive science capabilities.

Contents

1.0 INTRODUCTION

The overarching goal of the Predictive Science Academic Alliance Program (PSAAP) Center for Radiative Shock Hydrodynamics (CRASH) project [1] was to use scientific methods to assess and improve the predictive capability of a simulation code, based on a combination of physical and statistical analysis and experimental data. The specific focus of the project was radiative shocks, which develop when shock waves become so fast and hot that the radiation from the shocked matter dominates the energy transport. This in turn leads to changes in the shock structure. Radiative shocks are challenging to simulate because they include phenomena on a range of spatial and temporal scales and involve two types of nonlinear physics: hydrodynamics and radiation transport. Even so, the range of physics involved is narrow enough that one can reasonably aspire to model all of the physics with sufficient fidelity to reproduce the data.

To achieve its goal, the CRASH project relied on many computer simulations and a sequence of experiments that culminated in a final set of "year 5" experiments meant to assess the developed predictive capability. These experiments drove a radiative shock down an elliptical tube, while all of the experiments used to predict these year 5 experiments had involved only cylindrical tubes.

In this summary, we present overviews of the CRASH experiment, of the computer codes written to simulate the experiment, of the uncertainty quantification methods developed to combine experimental data and simulation results into a predictive capability, and of the solution verification techniques employed to assess the accuracy of simulated outputs.

2.0 THE CRASH EXPERIMENT

The physical phenomenon at the heart of the CRASH project is radiation hydrodynamics. Shocks in a fluid become radiative when they are strong enough to heat the fluid to temperatures at which the radiative energy flux is comparable to the incoming material energy. That is, there is a point at which a shock becomes strong enough that pure hydrodynamic considerations fail to predict the flow. As shown in Figure 1, hydrodynamic shock relations tell us that radiative energy flux, which depends on post-shock temperature, scales at a much faster rate with shock speed compared to the incoming material energy. If the shock structure were to not change, for fast enough shock speeds, the shock would eventually be radiating more energy than it had available from the incoming un-shocked flow! Therefore, to satisfy conservation of energy, the structure of the strong

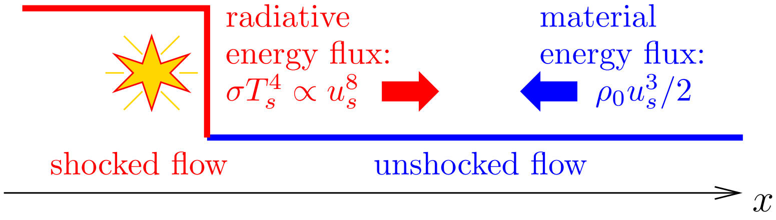
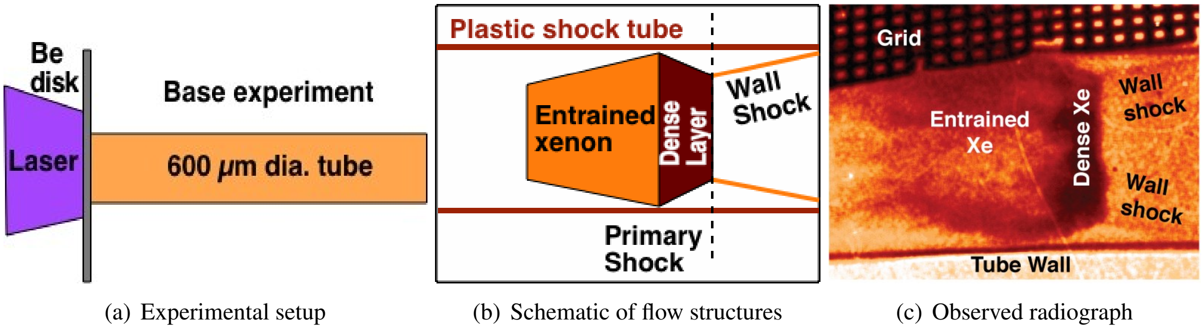


Figure 1: Energy balance in a shock. For low-to-moderate Mach number flows, the radiative energy flux is negligible. For strong shocks in which the shock speed u_s is sufficiently large (60 km/s in the CRASH system), the radiative energy flux is on the same order as the incoming material energy flux, which means that the unshocked flow "feels" the shock in the form of a radiative precursor before the hydrodynamic shock actually hits.

shock has to change. Predicting this structure in systems with multiple materials and non-trivial geometries is the science of interest in the CRASH project.

2.1 Overview

In the CRASH experiment, illustrated schematically in Figure 2(a), an approximately 3.8kJ laser of wavelength 0.35μm irradiates an approximately 20μm-thick beryllium disk for 1ns. This causes the beryllium to ablate and generate a shock that accelerates the formed plasma to over 100 km/s. The front of this plasma drives a shock through a xenon-filled polyimide (plastic) tube with an initial velocity of 200 km/s. After a shock of this strength, the dense xenon region is hot enough to radiate photons ahead of the shock [2] and preheat the unshocked xenon. Some of this radiation also heats the tube wall, which ablates the polyimide and creates a wall shock [3]. Figure 2(b) identifies the key features of the system, and Figure 2(c) shows a radiograph (obtained by measuring x-rays passing through the system) from one of the experiments.



(a) Experimental setup (b) Schematic of flow structures (c) Observed radiograph

Figure 2: Baseline (simple) experiment: a 1ns 3.8kJ laser irradiates a beryllium (Be) disk that ablates and sends a shock down a xenon (Xe) filled tube. Precursor radiation from the strong primary shock ablates the plastic wall of the tube causing oblique wall shocks. The "grid" in the radiograph is made of gold and serves as a fiducial for measuring the position of the shock.

Although the experiment is small in size, it requires an enormous amount of laser energy and setup time in the form of target construction and diagnostic preparation, making each shot an expensive undertaking. As a result, only $\mathcal{O}(10)$ shots could be taken each year of the CRASH project.

Whereas experiments performed in the first there years of the CRASH project were all with the baseline configuration in support of code/UQ development, calibration, and validation, the experiments performed in years four and five of the project were new. As illustrated in Figure 3, the geometry of the plastic tube was modified to contain a nozzle over the length of which the diameter decreased by about half: in the year 4 experiment the nozzle was isotropic/cylindrical, and in the year 5 experiment the nozzle was anisotropic/elliptical with a final aspect ratio of 2.

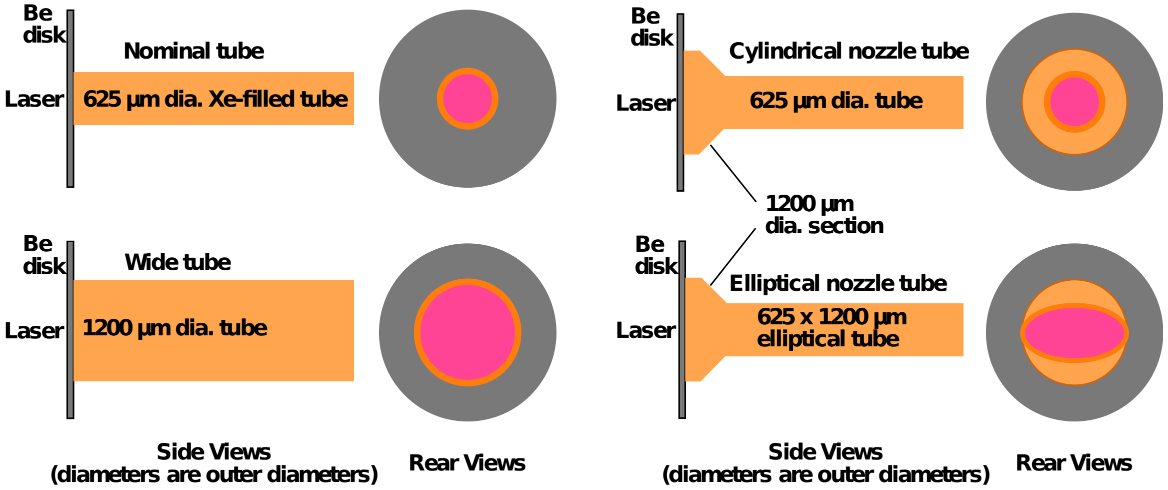


Figure 3: Schematic of the year 4 and 5 experiments in comparison to the baseline (nominal) experiments. The presence of a nozzle and the change of the tube cross-section to elliptical increase the complexity of the flow and make the task of prediction more challenging.

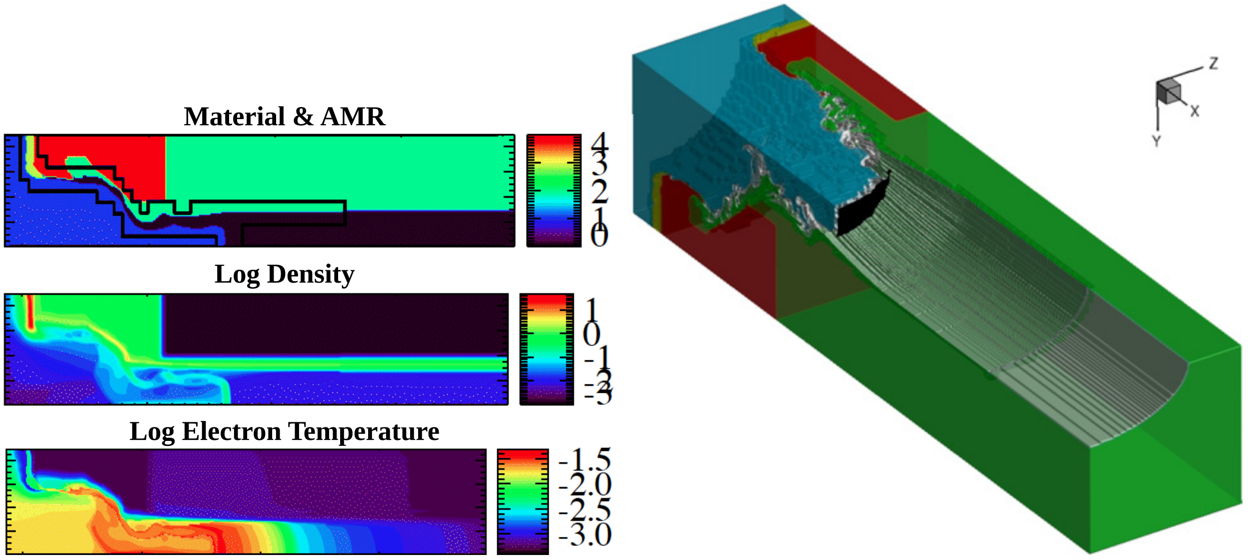
3.0 THE CRASH CODES

Several codes were written and used for the purpose of this project. Most significant among these was the CRASH code [4–6], which solved the radiation hydrodynamics equations in one, two, or three dimensions under various physics assumptions. Towards the end of the project, the CRASH code was outfitted with a laser modeling package, meaning that the experiment could be simulated with one code from the start of the laser shot. Prior to this, a separate laser deposition code, Hyades [7], was used in the first portion of the simulation, although the Lagrangian material treatment in this code made it prone to element "bow ties" in two dimensions, which had to be resolved manually and which prevented a large number of automated runs in support of the uncertainty quantification and prediction studies.

3.1 CRASH Capabilities

The CRASH code developed for the project evolved into a radiation-hydrodynamics solver with a laser package. It supports one, two, and three spatial dimensions, dynamic adaptive mesh refinement (AMR), level set treatment of multiple materials, self-consistent equations of state and opacities, multigroup-diffusion radiation transport, electron physics and flux-limited electron heat conduction, and a laser package based on ray tracing. Figure 4 shows sample solutions obtained from the CRASH code.

The CRASH code originated from a magnetohydrodynamics (MHD) solver used for space weather modeling at the University of Michigan, and it inherited that solver's Eulerian hydrodynamics treatment on parallel block-adaptive meshes. At its core, CRASH remains a finite volume solver with Riemann interface fluxes and linear reconstruction with limiting. Material interfaces are evolved using level sets, where each cell is treated



(a) Nominal tube: axisymmetric cut

(b) Elliptical nozzle: perspective view

Figure 4: Sample simulation results for the baseline (nominal) cylindrical tube and the year-5 elliptical nozzle, obtained using the CRASH code developed as part of the project. Colors in the figure on the right represent different materials, and the primary shock is highlighted in black.

as containing a single material. Each time step of the solver involves a mixed explicit/implicit update in which the diffusion and emission-absorption are evaluated implicitly via a preconditioned Newton-Krylov scheme.

3.2 Software Verification

As an extension of an existing large software framework, the CRASH code was programmed using best software verification practices developed as part of the framework. This means that: (1) the code was version controlled and maintained by a core team of developers; (2) new program functions and modules were implemented with *unit tests*, which were executed as part of nightly tests for CRASH and its parent code; (3) new code features/capabilities were implemented with *verification tests* that covered all aspects of the new feature, including restart, grid convergence studies, and model-to-model comparisons; (4) compatibility and reproducibility were checked nightly with compiles and runs on 9 different platforms/compilers on 1-4 cores; (5) weekly scaling tests on 128 and 256 cores revealed software implementation issues and confirmed result independence of the number of cores.

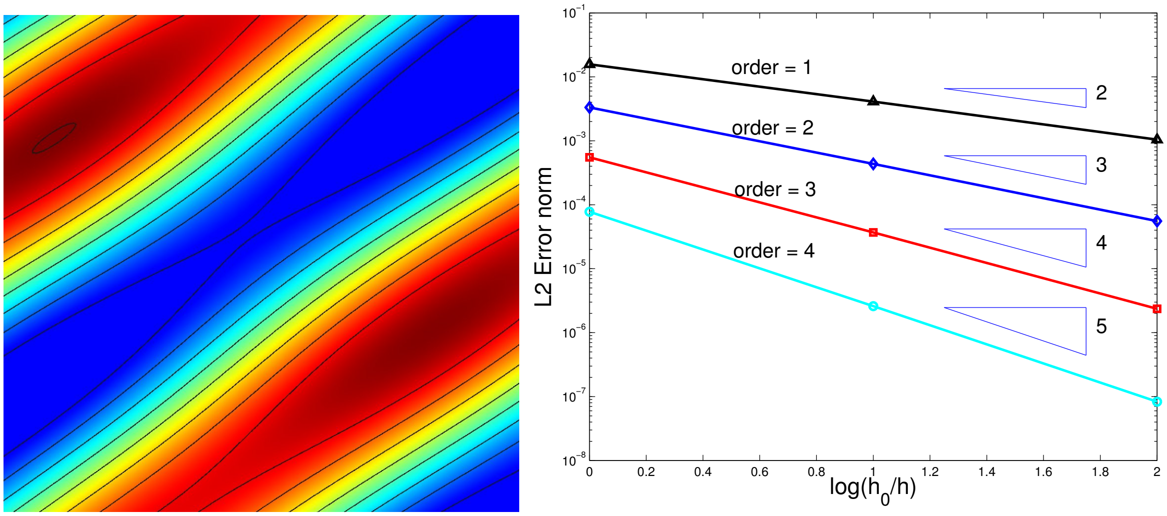
3.3 Manufactured Solutions

To verify if an implementation of a discretization is correct, the CRASH project used both analytical solutions and the method of manufactured solutions. The latter is necessary due to the difficulties in finding analytical test cases that fully stress the equations of radiative hydrodynamics. True to its name, this method lets us “make up” a solution, $\mathbf{u}^{\mathrm{MS}}$. Now, this made-up solution will generally not satisfy the original equations, PDE($\mathbf{u}$), but it will satisfy the altered equation

$$\mathrm{PDE}(\mathbf{u}) - \underbrace{\mathrm{PDE}(\mathbf{u}^{\mathrm{MS}})}_{\mathbf{s}^{\mathrm{MS}}} = \mathbf{0}. \tag{1}$$

So, to test our discretization of the PDE, we have to discretize an additional known source term, $\mathbf{s}^{\mathrm{MS}}$. Once we have this, we can run a simulation on an arbitrary domain with Dirichlet boundary conditions, $\mathbf{u}^{\mathrm{MS}}(\vec{x})$, and test how close the resulting solution is to the chosen exact solution. Based on interpolation theory, we expect the error between the discrete solution and $\mathbf{u}^{\mathrm{MS}}$ to converge as h^{p+1} for a characteristic mesh size h.

One simple manufactured solution equation is a sinusoidal variation of state components: $u_i = a_i + b_i \sin(c_i x + d_i y + e_i z + f_i t)$ for state component i. Figure 5 shows an example of using such a manufactured solution to test a discontinuous Galerkin finite element discretization of the gray-P1 radiation hydrodynamics equations. In this case, the problem is steady and two-dimensional, so $e_i = f_i = 0$. The other parameters a_i, b_i, c_i, d_i are $\mathcal{O}(1)$ and chosen randomly. In the convergence plots, we see that the continuous L_2 error norms in the discrete solutions decrease at the expected rates of $p + 1$. This demonstrates a successful verification of the discretization via a manufactured solution.



(a) Energy component of $\mathbf{u}^{\mathrm{MS}}$

(b) Error convergence

Figure 5: Manufactured solution tests for a discontinuous Galerkin finite element discretization of the steady two-dimensional gray-P1 radiation hydrodynamics equations, using spatial order p. Errors shown are continuous L_2 norms between the discrete solution and the exact manufactured solution, $\mathbf{u}^{\mathrm{MS}}$. Note that optimal order $p+1$ convergence rates are attained for all runs.

4.0 PREDICTIVE SCIENCE

The primary goal of the CRASH project was to demonstrate the capability of a numerical simulation code to predict radiative shock behavior in an unexplored region of the experimental space: ultimately the elliptical nozzle tube in the fifth-year experiment. The prediction was not blind to experimental data: experiments conducted in a different region of the input space were combined with simulations for the purpose of calibration. Towards this end goal, the CRASH project employed both sensitivity studies, to assess important parameters in the physical system, and predictive model construction, to assess the probability distribution functions of both physical parameters and experimental outputs. This section presents key developments and results in these areas.

4.1 Inputs, Parameters, and Outputs

In our uncertainty quantification studies, we distinguish between inputs, parameters, and outputs, as follows:

- $\mathbf{x}$ = vector of inputs, which are defined as quantities that are directly controllable or measurable in the experiments. These include quantities such as beryllium disk thickness, tube diameter, laser energy, xenon gas pressure, etc.

- $\boldsymbol{\theta}$ = vector of parameters, which are defined as quantities appearing in the equations used to model the physics of the experiments, but which are not directly controllable or measurable in the experiments. These include material properties such as opacities and specific heat ratios. We also refer to $\boldsymbol{\theta}$ as *calibration* parameters as some of these quantities must be inferred from experimental data in addition to prior information.

- $\mathbf{y}$ = vector of outputs, which are defined as quantities of interest measured in the experiments and in the simulations. Sample outputs include shock location, time of shock breakout from the beryllium disk, wall shock angles, amount of entrained xenon, etc.

The outputs $\mathbf{y}$ are the quantities that we would like to predict and that let us calibrate our parameters. Care must be exercised when defining these outputs. For example, an output defined as the point value of temperature at some location close to the shock might not be useful as it would be obscured in the experiments by noise in the radiograph, and in the numerical simulations it would be very sensitive to slight perturbations in the shock location. Because of the considerable variability in the flow and radiation structures in our system, both in physical experiments and in simulations, we developed a set of *integrated metrics* that are less sensitive to interface details. These metrics provide, in an integral form, the fundamental information given by the radiographs: how much dense xenon there is, where is it located, and how much is flowing near the edges. Within a fixed window, shown in Figure 6, we extract a set of metrics that are then calculated by post-processing pixels from the experimental or simulated radiographs.

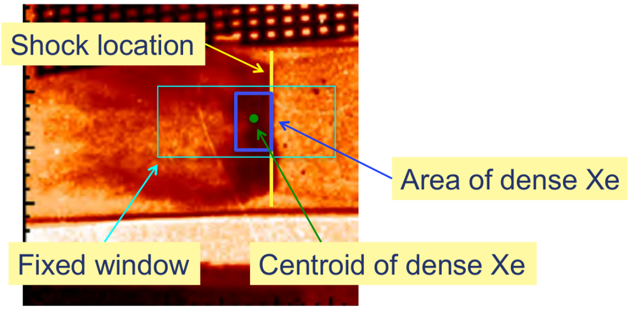


Figure 6: Examples of integrated metrics used as outputs in the experiments and simulations.

4.2 A Sensitivity Study

For much of the project duration, the CRASH code was initialized with outputs from the Hyades code. In order to have a calibrated initial condition modeling the laser irradiation phase of the experiment, we needed to assess the uncertainties associated with Hyades to gain a full understanding of the uncertainties in CRASH. One of the first uncertainty studies in the project was therefore of the Hyades code, specifically the one-dimensional version as this permitted automated runs with quick turnaround time. The results of this study provided evidence of the importance of different parameters within the code and were used to direct future studies.

In the sensitivity study, a 15-dimensional parameter space-filling Latin hypercube distribution was designed with a total of 512 runs with Hyades. The parameter list encompassed both inputs, such as the laser energy and the gas density, as well as physical and numerical code parameters, such as the xenon gamma (ratio of specific heats) and the mesh resolution. The experimental parameters were varied over a range defined by estimates of the variances from the experiments carried out at the Omega laser facility. The ranges of the code parameters were determined by expert-informed analysis of sensible ranges for each variable.

Field variables from ten locations in the simulated flows at 1.3 ns were collected into a large dataset used for sensitivity analysis and uncertainty quantification. The choice for the ten locations was motivated by what features would be important to develop a physics-informed emulator for the 1D code. This emulator was then used for the sensitivity study. Specifically, global sensitivities were evaluated by functional fitting of the field data with flexible regression methods (multivariate adaptive regression splines [8]) followed by random permutations of each input and computation of average RMS change over such permutations. Using this technique we were able to determine which inputs had the most significant effect on the response surface. These results are plotted in an influence plot in Figure 7 to show those inputs that have the largest global influence on the outputs. Results of the sensitivity study drew attention to several variables that deserved more attention. For example, after this study, we found from a literature review that the range used for the flux limiter variable was too large. Conversely, many of the other variables exhibiting low sensitivities (and for which the ranges were

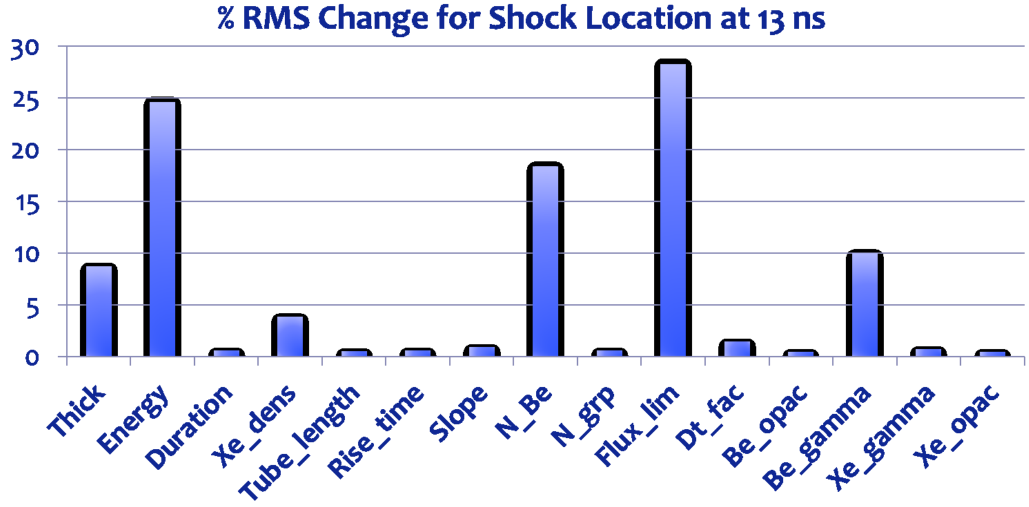


Figure 7: Influence plot from a study of the sensitivity of the shock location output to perturbations in fifteen different inputs, when using the one-dimensional Hyades code. The plot shows that the shock location is most strongly affected by three physical variables: beryllium thickness, gamma, and laser energy; and by two code variables: the number of zones used (N_{Be}) and the heat-conduction flux limiter.

deemed appropriate) could be neglected for later sensitivity and prediction studies.

4.3 Combining Data from Simulations and Experiments

Demonstrating predictive capability means making predictions in a new region of input space in which no prior experiments have been made. That is, using the simulation codes, we effectively extrapolate capabilities from one region of input space to another, ideally holding physics constant. To do this, we need to make use of all available information, which include simulations and experiments.

Our approach to combining numerical simulations and experimental data is based on that of Kennedy and O'Hagan [9], briefly reviewed here. Let y be the output of interest. First, suppose we have output values from n_s runs of our computer code, each with different inputs. We denote these simulation (s) data as $y_{s,j}$ and in anticipation of modeling the code, we write

$$y_{s,j} = \eta(\mathbf{x}_j, \mathbf{t}_j), \qquad j = 1 \ldots n_s. \tag{2}$$

In this expression, $\mathbf{x}_j$ and $\mathbf{t}_j$ are, respectively, the vectors of inputs and calibration parameters used in the j^{th} code run. We use $\mathbf{t}$ instead of $\boldsymbol{\theta}$ for the calibration parameters in order to reserve the notation $\boldsymbol{\theta}$ for the true values of the calibration parameters, which are assumed unknown. Now, suppose we also have n_f "field"/experimental measurements of the output. We write these as

$$y_{f,i} = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \epsilon_i, \qquad i = 1 \ldots n_f, \tag{3}$$

where $\delta(\cdot)$ is a model inadequacy or discrepancy function, and ϵ_i is the observation error in the i^{th} experimental measurement. Both $\eta(\cdot,\cdot)$ and $\delta(\cdot)$ are modeled as Gaussian processes, with a chosen covariance structure and prior distribution. Posterior distributions for the calibration parameters and outputs are then constructed by Markov-chain Monte-Carlo (MCMC) [10] sampling using a likelihood function of the form

$$L(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\rho}) \propto \boldsymbol{\Sigma}_y^{-1/2} \exp\left\{-(\mathbf{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y} - \boldsymbol{\mu})\right\}, \tag{4}$$

where $\mathbf{y} = (\mathbf{y}_s^T, \mathbf{y}_f^T)^T$ is a concatenation of the simulated and experimentally-observed output vectors, $\boldsymbol{\mu}$ is a constant mean function, and $\boldsymbol{\Sigma}_y$ is a covariance matrix that depends on the correlation parameters in $\boldsymbol{\rho}$.

4.4 Predicting Shock Location

In the CRASH experiment, the shock begins in the beryllium metal disk after the disk is irradiated by the laser, which is on for about 1ns. The shock breaks out of this disk about 400ps after the start of the laser pulse, and continues down the Xe-filled tube. One of our first predictive studies was to predict the location of the primary shock, using two of the codes, Hyades and CRASH. Hyades models the laser-plasma interaction and can predict the shock breakout time and the state of the system at 1.1 ns after the initiation of the laser pulse. The CRASH code, when initialized with this state at 1.1 ns, can predict the shock location at later times. The shock location can also be observed in experiments at observation times from 13ns to 26ns.

In the present case, we base the prediction on two experimental data sets: shock break-out time data and shock location data at 13ns, 14ns, and 16ns. Our interest is to predict the shock location at two later times, 20ns and 26ns. These predictions are then compared to experimental data to assess our capabilities in making the predictions. To do this, we define *two* models, as described next.

First, we construct a model for shock breakout time (BOT) that combines data from 1024 numerical simulations using the Hyades code with experimental measurements of the actual breakout time. This model is of the form described in Section 4.3, where $\mathbf{x}$ consists of four input variables and $\boldsymbol{\theta}$ consists of two calibration parameters. An MCMC sampling produces posterior distributions for the calibration parameters and for the parameters of the Gaussian-process models.

Looking at the posterior for the discrepancy function, $\delta(\mathbf{x})$, gives us some insight about the usefulness of our model. If the discrepancy function is significant relative to measurement error, we would say call our model fitting process "tuning" to the data. On the other hand, if the discrepancy is small, then we have more confidence that our model fitting is a form of calibration. Figure 8, on the right, shows a leave-one-out prediction study of the shock breakout time using our model. The magnitude of the discrepancy term for the breakout time model is small relative to the measurement error, and hence we proceed by using this model for calibration of the $\boldsymbol{\theta}$. Note, on the left, the figure also shows the results of a similar model constructed using the shock location at 13ns to 16ns as the data, and in this case the discrepancy is large relative to the measurement error meaning that we would not want to use this model for calibration.

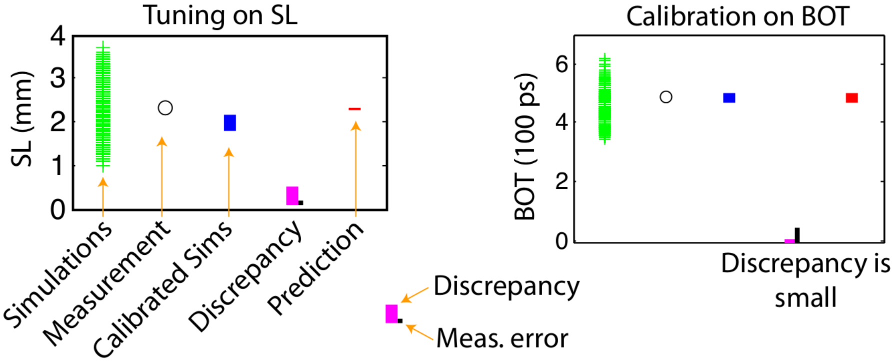


Figure 8: Tuning versus calibration for a shock prediction study. The full set of simulations (green), a single measurement to predict (open circle), calibrated code predictions (blue), discrepancy (pink) compared to measurement error (black), and the prediction of the measurement. All ranges shown are 95% prediction intervals.

After calibration on breakout time, we have posterior distributions for the two parameters in $\boldsymbol{\theta}$. We use these posteriors to construct another model, this time for the shock location at times of 16ns and less (where we have experimental data). Note, this does not include the shock location at 20ns or 26ns, which we are striving to predict. The model for the shock location is similar to that described in Section 4.3, except that $\boldsymbol{\theta}$ is no longer treated as a calibration parameter but instead a known "experimental" parameter drawn from the posterior distribution obtained from the previous breakout-time calibration step. In this model, the inputs $\mathbf{x}$ are also drawn from distributions that represent the known uncertainties in the experimental parameters. This second model is used to construct the Gaussian-process emulator and discrepancy, and the discrepancy can be analyzed to identify defects in the model. Since calibration occurs in a separate previous step, calibration of the $\boldsymbol{\theta}$ is not masking errors in the predicted shock location. The results show that this model tends to under-predict the shock location.

The final task is prediction. For this, we use the second model (for the shock location) to predict the shock at 20ns and 26ns. Since $\boldsymbol{\theta}$, and the Gaussian process modeling parameters ($\boldsymbol{\mu}, \boldsymbol{\rho}$) exist as posterior distributions, the prediction is inherently statistical. That is, we propagate the uncertainties to the shock location outputs at 20ns and 26ns through sampling. Figure 9 shows the result of this sampling. Note that the discrepancy term is relatively large because we are extrapolating to a new region of input space. The uncertainty in the emulator is smaller because there were simulation data in this region. Comparisons with actual experimental measurements at 20ns and 26ns show that even the smaller predictive interval from the emulator alone contains

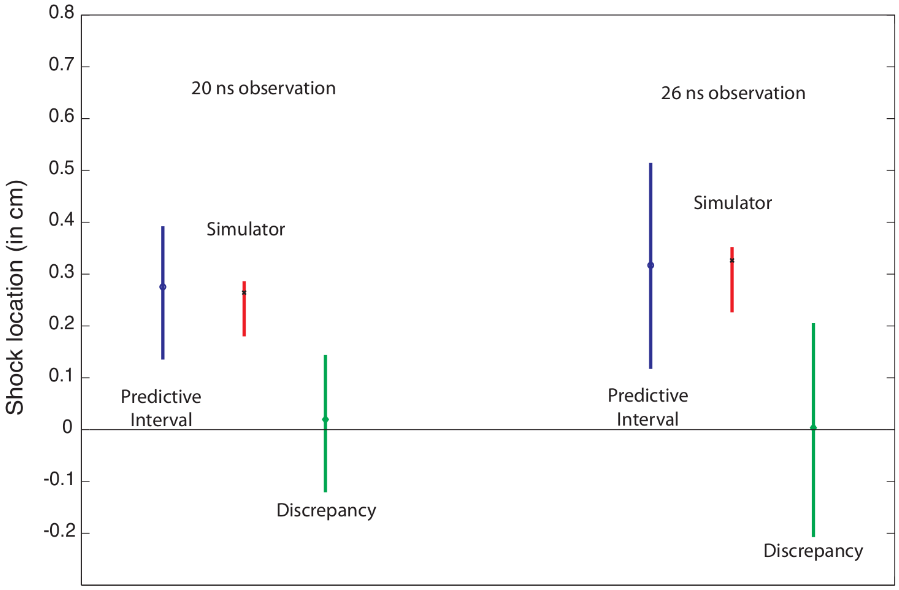


Figure 9: Results of applying the calibrated shock-location model for predicting new observations at 20ns and 26ns in advance of knowing the data. The red predictive interval is that due to propagating uncertainties in θ and x through the simulator, and the uncertainties in the parameters entering into the Gaussian-process emulator $\eta(\cdot,\cdot)$. In contrast, the blue predictive interval includes the uncertainty due to the discrepancy, which is large because it has been extrapolated from the 13-16ns range out to 20ns and 26ns.

the actual measurements. The median values from the predictions, 2750 microns at 20ns and 3200 microns at 26ns compare well with the experimental measurements of 2741 ± 70 microns and 3442 ± 30 microns, respectively.

4.5 A Multi-Fidelity Predictive Model

Section 4.3 presented an approach to combining simulations with experimental data, in the process calibrating parameters that could then be used for prediction. However, in the CRASH project, this modeling becomes more intricate because multiple simulation codes may be used. For example, the simulation could be done in one, two, or three dimensions, each of varying fidelity. In addition, the simulation codes could use different physics, so that the calibration parameters need not be the same between the codes. In this subsection we present a joint modeling approach developed to address this situation. Further details on this approach can be found in [11].

Consider a case where we have two simulation codes, 1 and 2, where code 1 is of lower fidelity than code 2. This usually means we can afford to run more simulations of code 1 but that the runs of code 2 will be more accurate. The codes share a vector of calibration parameters $\boldsymbol{\theta}$, but in addition to these, code 1 also uses a vector $\boldsymbol{\theta}_1$ of its own calibration parameters, while code 2 uses a vector $\boldsymbol{\theta}_2$ of its own calibration parameters. Our approach to multi-fidelity modeling is to calibrate the first computer code to the second one (of higher fidelity) and then to model the experimental observations using the simulator of higher fidelity. That is, we write:

$$\text{Low-fidelity code outputs:} \quad y_{1,j} \quad = \quad \eta_1(\mathbf{x}_j, \mathbf{t}_j, \mathbf{t}_{1,j}), \qquad j = 1 \ldots n_1, \tag{5}$$

$$\text{High-fidelity code outputs:} \quad y_{2,k} \quad = \quad \eta_2(\mathbf{x}_k, \mathbf{t}_k, \mathbf{t}_{2,k}), \qquad k = 1 \ldots n_2, \tag{6}$$

where again we use $\mathbf{t}$ in place of $\boldsymbol{\theta}$ to distinguish between code inputs and true values of the calibration parameters. Calibrating the low-fidelity code to the high-fidelity code means writing outputs of the latter as a discrepancy-adjusted version of outputs of the former,

$$y_{2,k} = \eta_1(\mathbf{x}_k, \mathbf{t}_k, \boldsymbol{\theta}_1) + \delta_{1\to 2}(\mathbf{x}_k, \mathbf{t}_k, \mathbf{t}_{2,k}), \qquad k = 1 \ldots n_2 \tag{7}$$

where $\delta_{1\to 2}(\cdot,\cdot,\cdot)$ represents the systematic difference between the partially-calibrated low-fidelity model and the high-fidelity code. This discrepancy term includes $\mathbf{t}_k$ and $\mathbf{t}_{2,k}$ because these can be modified in the high-fidelity code. Note the use of $\boldsymbol{\theta}_1$ in the first term amounts to partially calibrating the model of the first computer code to outputs of the second computer code. We now model the experimental measurements using the high-fidelity code,

$$y_{f,i} = \eta_2(\mathbf{x}_i, \boldsymbol{\theta}_i, \boldsymbol{\theta}_2) + \delta(\mathbf{x}_i) + \epsilon_i, \qquad i = 1 \ldots n_f. \tag{8}$$

The next step is similar to the single-code case: we model the code outputs, $\eta_1(\cdot,\cdot,\cdot)$ and $\eta_2(\cdot,\cdot,\cdot)$, and the discrepancies $\delta_{1\to 2}(\cdot,\cdot,\cdot)$ and $\delta(\cdot)$, as Gaussian processes with parametrized covariance structures. We then take many MCMC samples to jointly fit all of the outputs (low/high fidelity codes and the experiments) by varying all of the inputs and calibration parameters. As priors, we use inverted gamma distributions for the variance components (appearing in front of the covariance expressions), beta distributions for the correlation parameters, and log-normal distributions for the calibration parameters.

4.6 Predicting Shock Breakout Time

Consider again the shock break-out time discussed in Section 4.4. Suppose we want to construct a predictive model for this breakout time using two simulation codes: (1) a one-dimensional version of CRASH; and (2) a two-dimensional version of CRASH. The latter is able to resolve more processes and interactions and hence is expected to be more accurate. Of course, it is also more expensive.

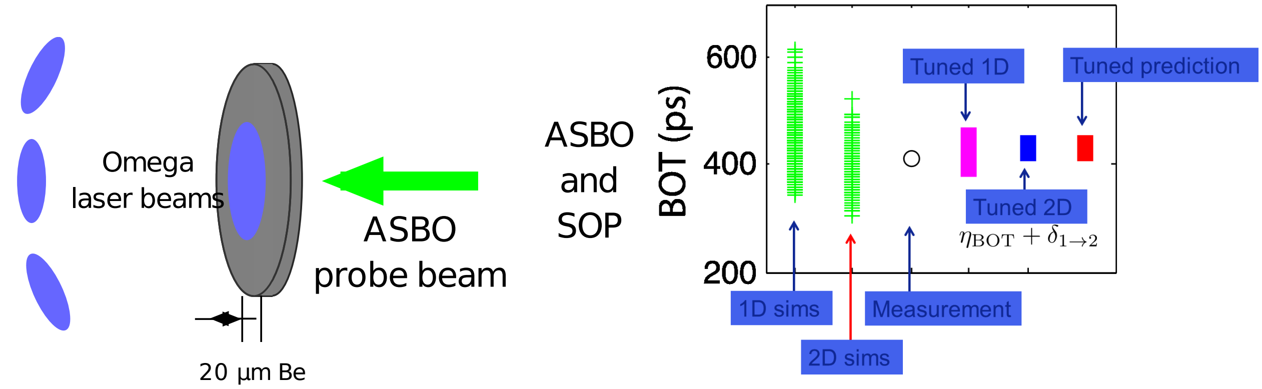
Two design/input variables are used in this prediction study: the beryllium disk thickness, x_1, and the laser energy, x_2. The electron flux limiter is the common calibration parameter; the 1D-CRASH code also requires a laser energy scale factor (that 2D-CRASH does not), while the 2D-CRASH code also requires the beryllium ratio of specific heats and a wall opacity scale factor (not required by 1D-CRASH). All inputs are scaled to the unit interval before the fitting.

We have $n_1 = 365$ simulations of 1D-CRASH, $n_2 = 104$ simulations of 2D-CRASH, and 8 experimental measurements of the break-out time. Using the joint calibration model discussed in the previous subsection, 10,000 MCMC simulations (minus the first 2000 to remove burn-in) yield posterior distributions of the calibration parameters. To test the predictive capability of the calibrated joint model, we perform a leave-one-out study in which we remove one of the experimental observations, construct the model, and use this model to predict the removed observation. Figure 10 shows the result of this study. The tuned prediction range (95% confidence interval) contains the measurement in a reasonably tight range.

4.7 Predicting the Fifth Year Experiment

As shown in Figure 3, the fifth year experiment consisted of an elliptical tube with a nozzle connecting the tube to a circular initial section. Predicting outputs such as shock location in this geometry was one of the critical aspects of the predictive capability assessment.

The statistical model for this predictive task is again constructed using simulations and experiments, The experiments used are those from the 2008 and 2010 experiments. All of the 2008 experiments, and all but three



(a) Measuring shock breakout time in the experiments

(b) Predictive results

Figure 10: Prediction of shock breakout time using experimental data and simulations at two different levels of fidelity (1D-CRASH and 2D-CRASH).

of the 2010 experiments, were performed on circular shock tubes with a small diameter (approximately 575 microns). Three of the 2010 experiments used circular tubes with a diameter of 1150 microns. The 2012, fifth year, experiments to be predicted have an elliptical shaped tube and a nozzle at the front of the tube. We had two observations with a nozzle on circular tubes, but none with an elliptical tube. Thus, there are no observations in the regime (elliptical tube) where the predictions are to be made. The defining feature that captures the shape of the tube is the *aspect ratio*, defined as the ratio of the major-axis diameter to the minor-axis diameter.

The code runs arise from two separate simulation suites - run-sets 12 and 13 (RS12 and RS13). Run set 12 consists of 128 CRASH-2D simulations of the baseline experiment. Run set 13 consists of 80 CRASH-3D simulations on the full year-5 experimental geometry. The inputs for each run set and the ranges explored are shown in Table 1. Notice that several of the inputs are held constant in RS12 and RS13. In particular, the energy flux limiter is held constant in RS13 because the outputs were found to be relatively insensitive to this input in earlier studies, so that the only calibration parameter to be estimated is the energy scale factor. The inputs for a sample of 428 simulations from RS12 and RS13, the 2008 and 2010 experiments, and the fifth year experiments are shown in Figure 11. These plots show that the 2008 and 2010 experiments (red circles) fall within the input region covered by the simulations. However, the fifth year experiment (light blue circles) is clearly an extrapolation in the aspect ratio. That is, the aspect ratio of the experiments used for the model construction is equal to one, but the aspect ratio in the fifth year prediction experiment is two.

We construct a predictive model using the approach outlined in Section 4.3. To gain insight into which inputs $\mathbf{x}$ and parameters $\boldsymbol{\theta}$ are important in the Gaussian-process emulator, $\eta(\mathbf{x}, \boldsymbol{\theta})$, we visualize the emulator response as a function of each individual input/parameter by marginalizing over the other inputs/parameters. This is shown in Figure 12. Note that the output is fairly insensitive to several of the inputs governing the tube geometry, which the exception of tube diameter which is quite active. In general, the model appears insensitive to outputs in which no extrapolation is required.

Once the model is available, we sample outputs for the predictive case using the following procedure:

1. Sample the inputs $\mathbf{x}$ using prior information and the calibration parameters for the statistical model using the posteriors available from the constructed models.

2. Use the sampled inputs and parameters to evaluate the Gaussian-process emulator output, $\eta(\mathbf{x}, \boldsymbol{\theta})$.

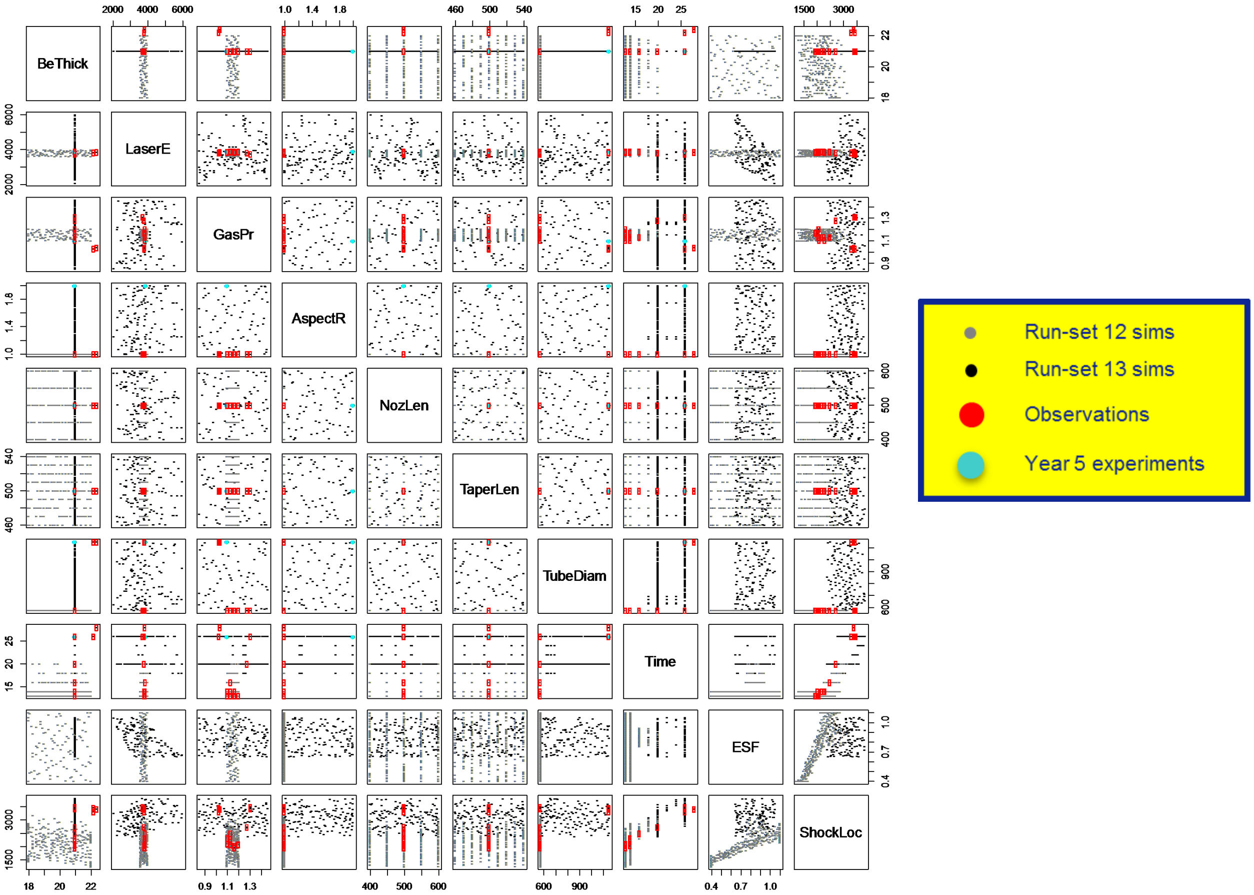


Figure 11: Pairwise scatter plots of the inputs (8 values in x) calibration parameter, one θ, and output value, y = shock location at a specific time, from data used for constructing the predictive model of the year-5 experiment.

Table 1: Inputs and associated distributions for run sets 12 and 13, used for the construction of the model in the year-5 experiment prediction.

Design variables		
Input	**RS12**	**RS13**
Be thickness (microns)	[18,22]	21
Laser energy (J)	[3600,3990]	
Effective laser energy (J)*		[2156.4,4060]
Xe fill pressure (atm)	[1.100,1.2032]	[0.852,1.46]
Tube diameter (microns)	575	[575,1150]
Taper length (microns)	500	[460,540]
Nozzle length (microns)	500	[400,600]
Aspect ratio (microns)	1	[1,2]
Calibration parameters		
Input	**RS12**	**RS13**
Electron flux limiter	[0.04, 0.10]	0.06
Energy scale factor	[0.40,1.10]	[0.60,1.00]

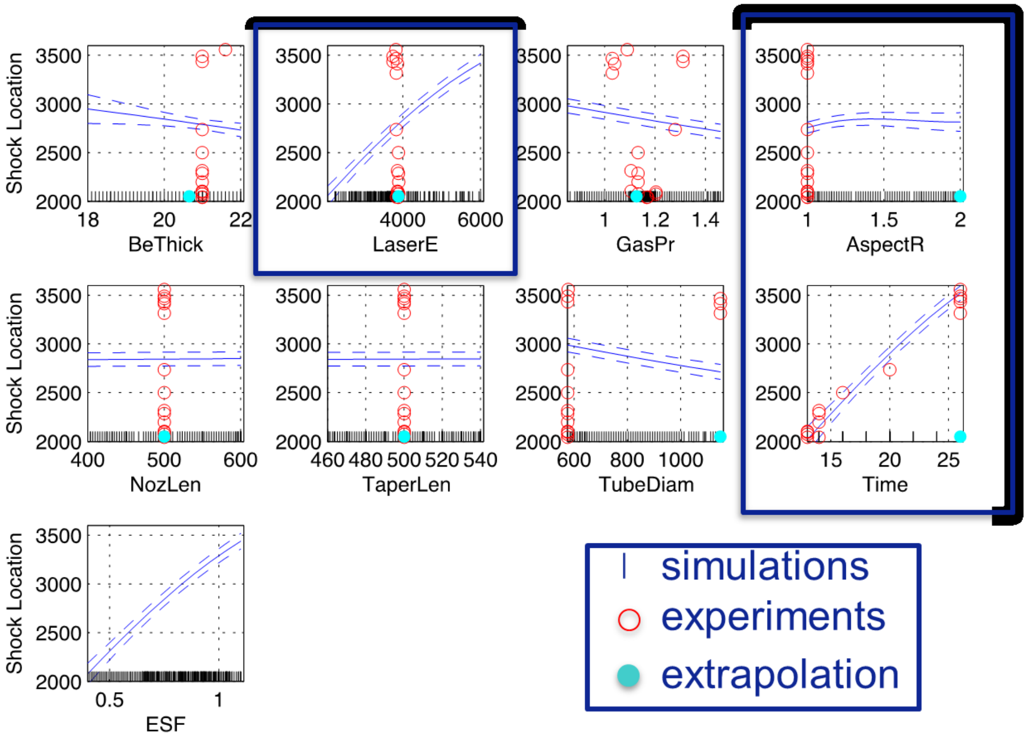


Figure 12: Marginalized sensitivities of the Gaussian-process emulator, $\eta(\mathrm{x}, \theta)$, of the simulations, calibrated for predicting the fifth-year experiment.

3. Use the sampled inputs to estimate the discrepancy, $\delta(\mathbf{x})$, and add this to the prediction.

4. Sample a value for the observation error and add it to the prediction.

These steps are repeated many times in order to build up a posterior distribution for the output of interest, $\mathbf{y}$, which is the shock location at a fixed time. We can also look at the entire shock trajectories versus time, and Figure 13 shows some of the sampled trajectories together with 95% prediction intervals and the experimental measurements (that went into building the predictive model). Note that each of the experimental measurements lies within the 95% prediction intervals, which lends come credibility to the predictive model.

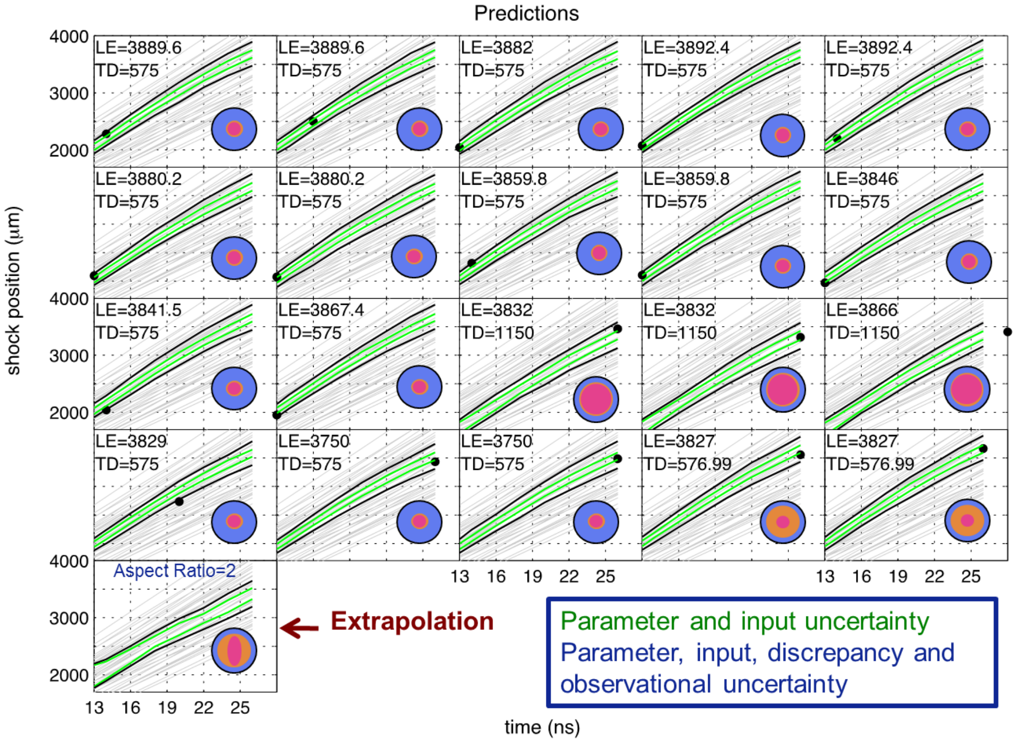


Figure 13: Posterior prediction intervals at the nominal settings for shock location versus time viewed along the minor axis, obtained from the model constructed to predict the year-5 experiment. Gray curves show a sample of predicted trajectories at different input settings, green curves are 95% prediction intervals incorporating parameter and input uncertainty, black curves form 95% prediction intervals incorporating parameter and input uncertainty, discrepancy variation and observational uncertainty, and the circles are the experimental observations from the 2008 and 2010 experiments.

The last frame in Figure 13 shows the predicted trajectories for the year-5 experiment, which is an extrapolation in the aspect ratio. The same trajectories are repeated in Figure 14, together with the actual year 5 experimental measurements. This figure shows that the model slightly under-predicts the shock location. In other words, the majority of the shock location observations in the experiments lie above the prediction intervals specified in our constructed model. We did not obtain a conclusive explanation for this under-prediction, but our best hypothesis is that this is due to our diffusive radiation transport model getting the wall heating wrong in the case of a tube with an elliptical cross-section, due to a difference in the solid angle along the major and minor axes. That is, our predictive capability relies on the physics “carrying-over” from the calibration experiments to the prediction experiments, but in this case a model (diffusive radiation transport) that was appropriate for

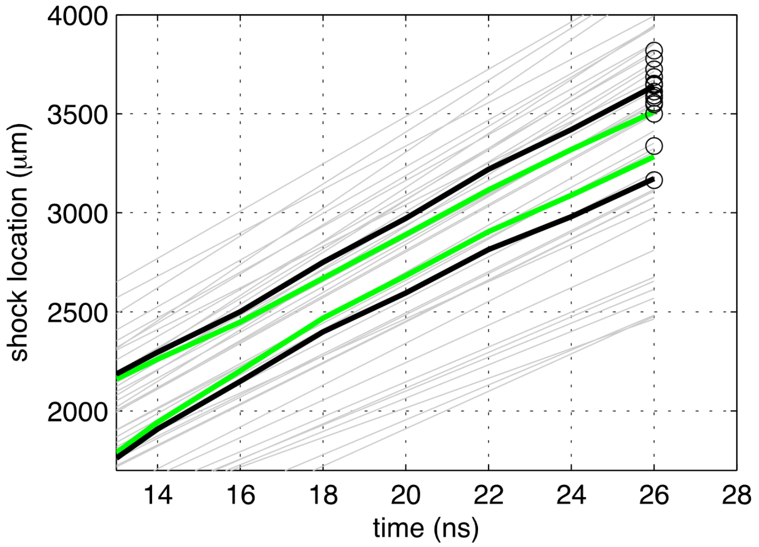


Figure 14: Posterior prediction intervals and actual measurements for the year 5 experiment. The color code for the simulated shock trajectories is the same as in Figure 13.

the circular tube may not have been as applicable to the elliptical tube. Nevertheless, the predictive model still does a decent job: differences in the prediction are roughly on the order of the experimental variability.

5.0 SOLUTION VERIFICATION AND ADAPTATION

Numerical errors due to insufficient spatial and temporal mesh resolution can affect results, often in seemingly subtle but significant ways. The latter point is especially true for convection-dominated flows, which include radiation hydrodynamics. One of our goals in the CRASH project was to quantify the effect of these numerical errors and to reduce them through mesh adaptation. This section describes some of our work in this regard.

5.1 CRASH Code Convergence

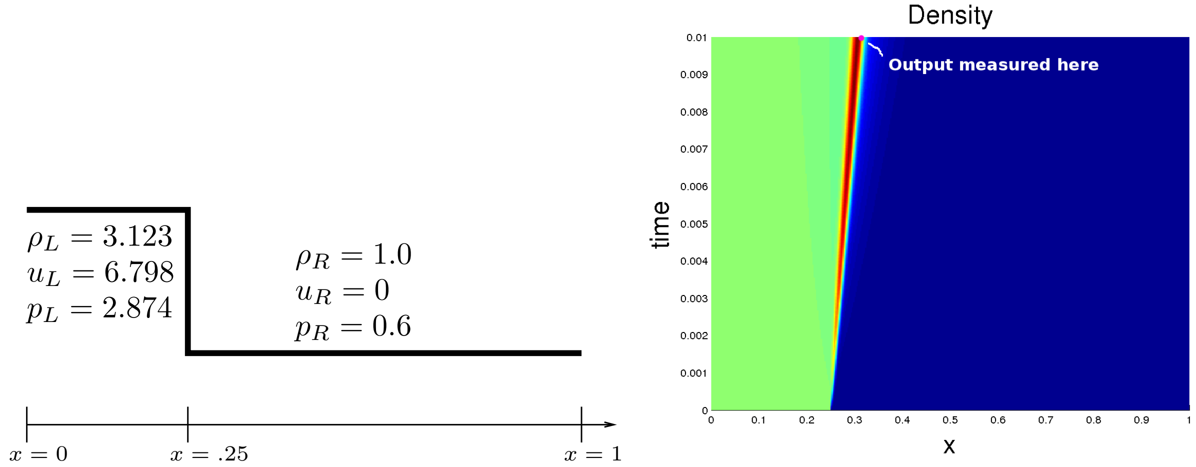
In this subsection we summarize solution verification efforts using the CRASH code, which consist primarily of grid refinement convergence studies.

5.1.1 A Shock-Tube Problem

As a first convergence study, we consider the one-dimensional shock-tube problem illustrated in Figure 15. The governing equations are compressible Euler with a gray-P1 radiation transport model [12],

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{u}) + \mathbf{S}(\mathbf{u}) = \mathbf{0}, \tag{9}$$

where the state vector is $\mathbf{u} = \left[\rho, \rho\vec{v}, \rho E, E_r, \vec{F}_r\right]^T$, the flux is $\vec{\mathbf{F}} = \left[\rho\vec{v}, \rho\vec{v}\vec{v} + p\underline{\delta}, \rho\vec{v}H, \vec{F}_r, \frac{1}{3}E_r\underline{\delta}\right]^T$, and the source is $\mathbf{S} = \left[0, \vec{S}_m, S, -S, -\vec{S}_m\right]^T$, with $\vec{S}_m = -\sigma_t\left(-\vec{\psi} + \phi\frac{\vec{v}}{c}\right)$, $S = -\sigma_t c\left(\phi + \vec{\psi}\cdot\frac{\vec{v}}{c}\right)$, $\phi = a_r T^4 - E_r$,



(a) Initial condition

(b) Space-time contours of density ($\sigma_t = 100$)

Figure 15: Setup for a one-dimensional shock tube problem in which we apply output-based error estimation to predict a scalar point-density output at the end of a certain time horizon.

and $\vec{\psi} = \frac{\vec{F}_r}{c} - \frac{4}{3}E_r\frac{\vec{v}}{c}$. Since the problem is one-dimensional the spatial vectors (e.g. $\vec{v}$) become scalars.

In the present shock tube simulation, E_r and $\vec{F}_r$ are initially set to their equilibrium values, $E_r = a_r T^4$, $\vec{F}_r = \frac{4}{3}E_r\vec{v}$, and the remaining parameters are $\gamma = \frac{5}{3}$, $a_r = 44.93$, $c = 100$. The time range for the simulation is $t \in [0, 0.01]$, and the output of interest is the density at the final time and at one point in space, $x = .30825$ chosen to be inside the shock structure.

The shock-tube problem was run using the CRASH code for the case $\sigma_t = 0$. Figure 16 shows how the density profile changes with spatial mesh resolution. Note that the time step changes proportionately with the mesh spacing to maintain a constant Courant-Friedrichs-Lewy (CFL) number.

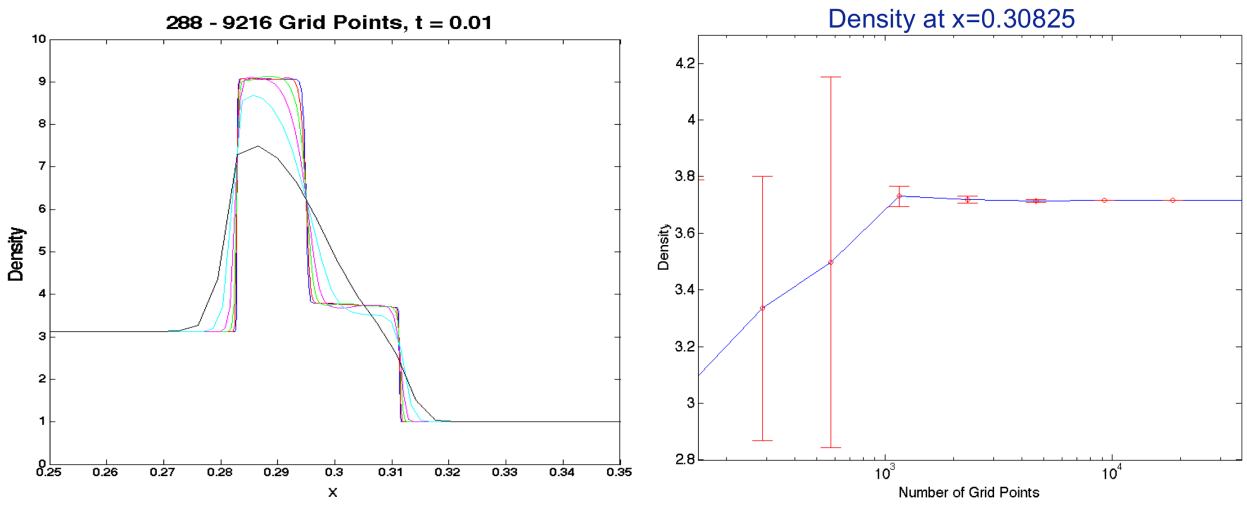
The error bars in Figure 16 come from a "grid convergence index" (GCI) calculation [13], which is based on an asymptotic regime assumption. Specifically, if the solution is in an asymptotic regime, we expect outputs to converge as

$$y_h = y_{\text{exact}} + Ch^p, \tag{10}$$

where h is the mesh spacing, y_h is the output on a discretization with spacing h, C is a problem/output dependent constant, and p is the order of convergence (assumed or measured with Richardson extrapolation). Given solutions at two refinement levels, h and H, we can use (10) to eliminate C in an expression for the output error,

$$y_h - y_{\text{exact}} = Ch^p = Ch^p\frac{h^p - H^p}{h^p - H^p} = \frac{Ch^p - CH^p}{1 - (H/h)^p} = \frac{y_h - y_H}{1 - (H/h)^p} \equiv \text{GCI}\frac{y_h}{F_s}. \tag{11}$$

That is, the grid convergence index is a measure of the solution error relative to y_h and includes a "factor of safety" F_s (typically 1-3) intended to mitigate the risk of underestimating the error. Error bars can then be drawn at $\pm\text{GCI}y_h$ around the output value y_h. As shown in Figure 16 the error bars decrease with mesh refinement (although this decrease is not always monotonic) indicating an approach to a converged solution.



(a) Density profiles

(b) Point density output convergence

Figure 16: Density profiles and point output convergence for various spatial mesh resolutions in a CRASH simulation of the one-dimensional shock-tube problem. Error bars in the figure on the right are estimates obtained using a grid convergence index (GCI) with $F_s = 3$.

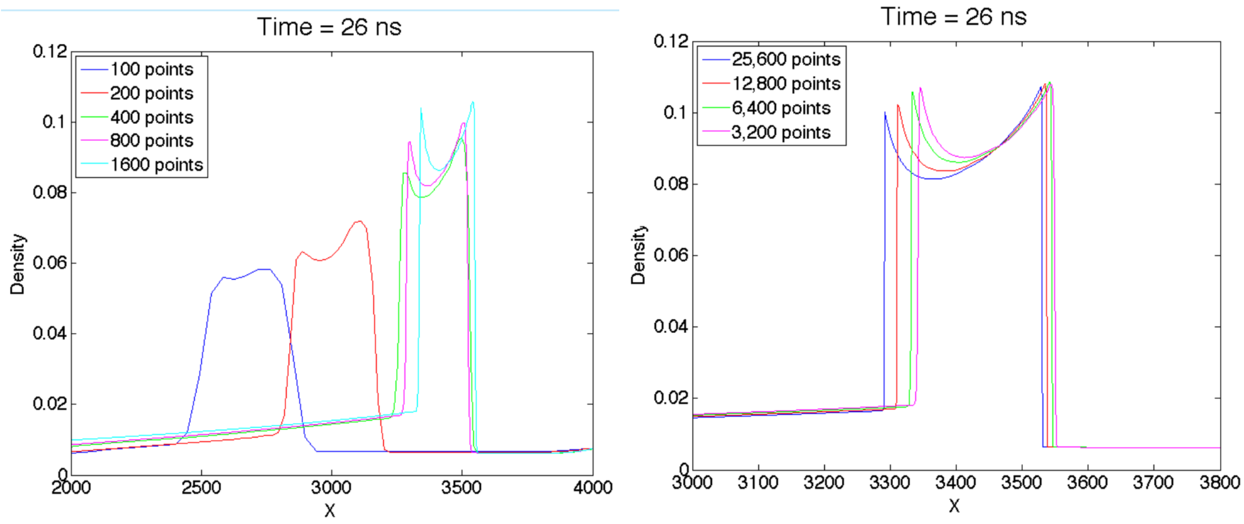
5.1.2 The CRASH Problem

Turning now to the one-dimensional CRASH problem (laser/beryllium driven shock in xenon), we perform a similar grid convergence study. Figure 17 shows how the density profile at a time of 26ns changes with grid resolution. Note the non-monotonic behavior of the density peaks with respect to grid resolution.

Finally, we present the results of a convergence study for the two-dimensional CRASH problem. Figure 19 shows the density field variable at one simulation time snapshot for a sequence of mesh refinements. As shown, increased mesh resolution uncovers new flow structures and significant changes occur even between the last two mesh resolutions, which are already at a refinement level that could not be extended to three dimensions. This lack of convergence is likely caused by a combination of the same issues preventing one-dimensional convergence and multi-dimensional hydrodynamic instabilities due to the use of an inviscid (compressible Euler) model for a system with nonzero physical viscosity. Further work is therefore required in the physical modeling aspects of this problem to observe convergence. Nevertheless, the variations between the shock location results on the different meshes are still quite small: 100μm is below the experimental variability.

5.2 Output-Based Error Estimation and Mesh Adaptation

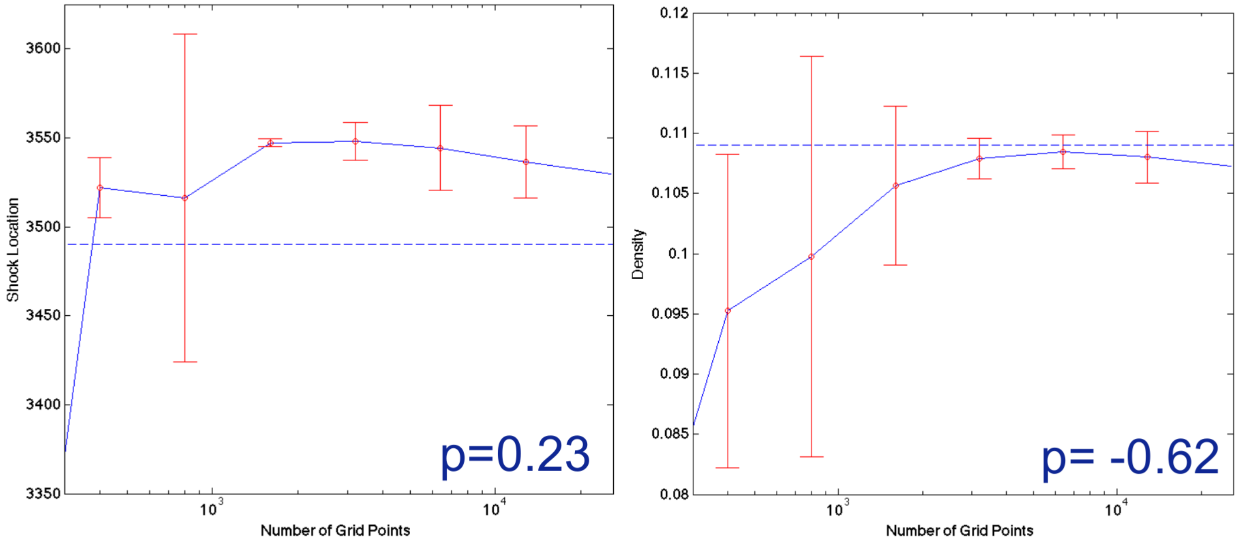
On a parallel track with the core CRASH code effort, we investigated the applicability of output-based error estimation and adaptation [14] methods to unsteady radiation hydrodynamics solutions. These methods not only quantify the numerical error in scalar outputs of interest but also drive adaptive refinement of both the spatial and the temporal discretizations, with the goal of improving output accuracy. In other applications, in particular external aerodynamics [15, 16], we have observed large benefits in total degrees of freedom and computational time when using output-based adaptive indicators compared to uniform refinement and to heuristic indicators. The implication is that in these problems, resolution over the entire domain is not necessary to predict the desired quantities, and that the identification of the important regions of the domain is not straightforward. In



(a) Low mesh resolution

(b) High mesh resolution

Figure 17: Convergence study of the one-dimensional CRASH problem, using the CRASH code. Density profiles for five lowest grid resolutions show a different trend (density peak moves to right and increases in height with mesh resolution) compared to the density profiles for the four highest grid resolutions (density peak moves to left and decreases in height with mesh resolution).



(a) Shock position

(b) Post-shock density

Figure 18: Convergence of shock location and post-shock density in the one-dimensional CRASH problem, using the CRASH code. Monotonic convergence with respect to mesh resolution is not observed, and the outputs continue to vary even at the finest meshes which have over 10,000 mesh points. The reason for the lack of convergence is not precisely known, but several factors could be involved: the multigroup radiation model, the material level sets, and/or discontinuities due to strong radiative shocks.

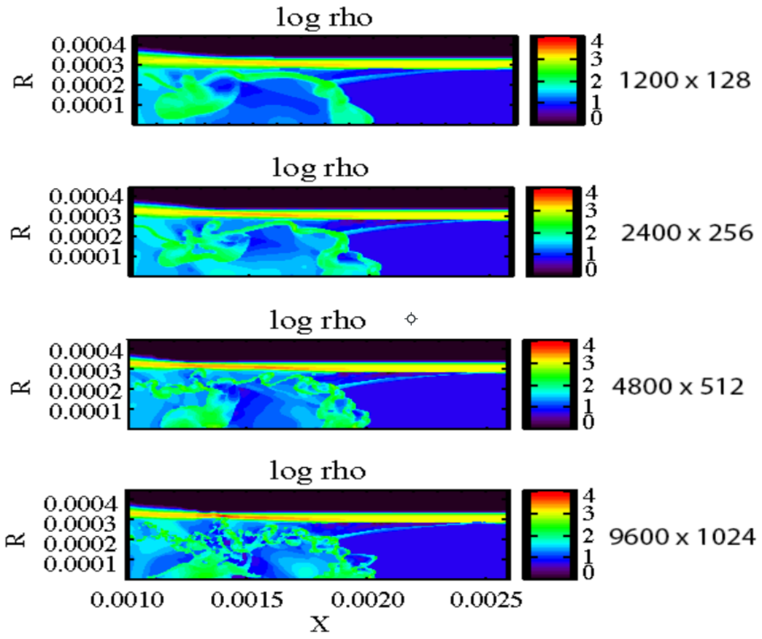


Figure 19: A visual assessment of the convergence of the density field variable under uniform mesh refinement for the two-dimensional CRASH problem. The shock location and morphology change significantly with grid resolution (e.g $50\mu m$ for the shock location), and these changes are not monotonic.

this section we show a similar potential exists for radiation hydrodynamics simulations.

The target problem for the output-based adaptation studies is still the one-dimensional shock-tube presented in Section 5.1.1, and the equations remain compressible Euler with a gray-P1 radiation treatment. One difference is that we now set $\sigma_t = 100$, which has the effect of smoothing out the density profile. The time range for the simulation remains $t \in [0, 0.01]$, and the output of interest remains the density at the final time and at $x = .30825$, which is still inside the shock structure.

Instead of using the CRASH code, we now solve (9) with a discontinuous Galerkin finite element method of order p in space and order r in time. $p = 0$ is used to improve robustness in the vicinity of the strong shocks, while $r = 1$ is used for the temporal order. The discrete residuals associated with the space-time discretization can be written as

$$\mathbf{R}(\mathbf{U}, \mathbf{x}) = \mathbf{0}, \tag{12}$$

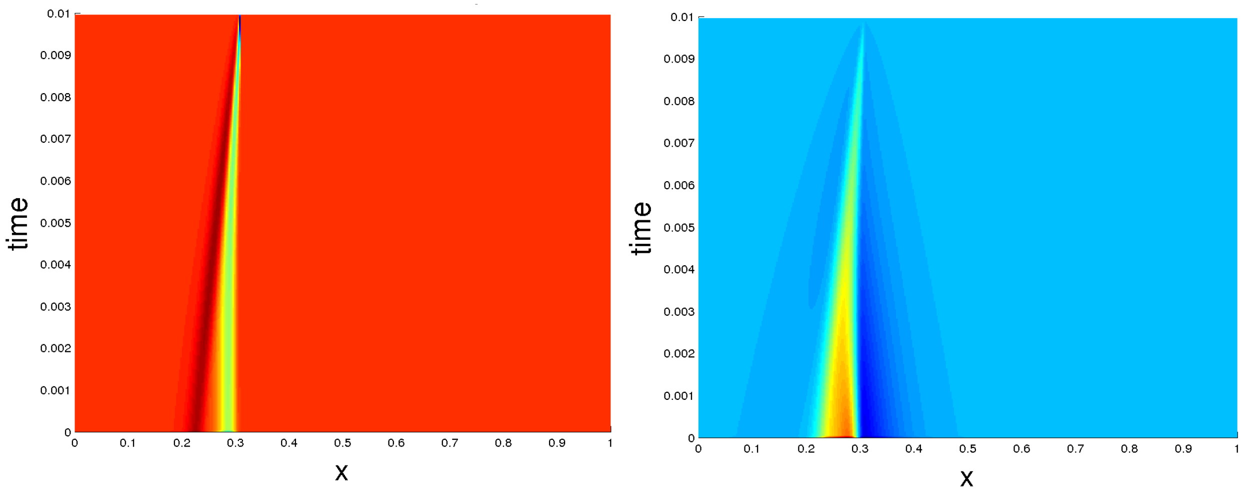
where $\mathbf{U} \in \mathbb{R}^N$ is the state vector over all of the time nodes, $\mathbf{R} \in \mathbb{R}^N$ is the unsteady residual vector at all of the time nodes, N is the total number of unknowns in space and time, and $\mathbf{x}$ is a vector of input parameters. Note, in a finite element method in time, “time nodes” actually correspond to temporal basis functions, for which the states are the coefficients and from which the residuals are obtained via the method of weighted residuals.

5.2.1 The Discrete Adjoint

We are interested in a scalar output, y, computed from the solution to the PDE,

$$y \quad = \quad y(\mathbf{U}). \tag{13}$$

The discrete adjoint, $\mathbf{\Psi} \in \mathbb{R}^N$, is a vector of sensitivities of the output to the N residuals. That is, each entry of the adjoint tells us the effect that a perturbation in the same entry in the residual vector would have on the output y. Figure 20 provides an illustration of the adjoint solution for the present shock-tube problem.



(a) Conservation of mass adjoint

(b) Conservation of radiation energy adjoint

Figure 20: Two components of the adjoint solution for the point output targeted in the unsteady, one-dimensional, radiation-hydrodynamics shock-tube problem. Note the opposite temporal trend as compared to the primal solution: the adjoint spreads out for earlier simulation times, as it indicates locations at which residual sources affect the output.

One common source of residual perturbations is changes in input parameters for a problem, and so we ground the adjoint presentation in the context of a local sensitivity analysis. Consider a situation in which p_x parameters, $\mathbf{x} \in \mathbb{R}^{p_x}$, affect the PDE according to the following chain of dependence,

$$\mathbf{x} \rightarrow \mathbf{R}(\mathbf{U}, \mathbf{x}) = \mathbf{0} \rightarrow \mathbf{U} \rightarrow y(\mathbf{U}). \tag{14}$$

We are interested in how y changes (locally for nonlinear problems) with $\mathbf{x}$,

$$\frac{dy}{d\mathbf{x}} \in \mathbb{R}^{1\times p_x} = p_x \text{ sensitivities.} \tag{15}$$

Note, if y depends directly on $\mathbf{x}$ we would add $\frac{\partial y}{\partial \mathbf{x}}$ to the above sensitivity, but for clarity of presentation we consider only the case when $y = y(\mathbf{U})$. To compute the above sensitivities, we could use finite differencing or forward linearization, but these calculations become expensive when p_x is moderate or large because of the need to re-solve the system for each parameter. An alternative is the adjoint approach, which requires an inexpensive residual perturbation calculation followed by an adjoint weighting to compute the effect on the output. That is, we write

$$\frac{dy}{d\mathbf{x}} = \mathbf{\Psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{16}$$

This approach is efficient for computing a large number of sensitivities for one output, as the cost after computing the output adjoint is one residual perturbation calculation and one vector product per sensitivity.

To derive an equation for the adjoint, we consider adding a residual perturbation, $\delta\mathbf{R} = \frac{\partial\mathbf{R}}{\partial\mathbf{x}}\delta\mathbf{x}$ to the left-hand side of (12). Linearizing the equation, we can solve for the corresponding state perturbation, $\delta\mathbf{U} = -\left[\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right]^{-1}\delta\mathbf{R}$. Also linearizing the output, we now have

$$\delta y = \frac{\partial y}{\partial\mathbf{U}}\delta\mathbf{U} = \underbrace{-\frac{\partial y}{\partial\mathbf{U}}\left[\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right]^{-1}}_{\mathbf{\Psi}^T \in \mathbb{R}^N}\delta\mathbf{R}. \tag{17}$$

Taking the transpose of the equation $\mathbf{\Psi}^T = -\frac{\partial y}{\partial\mathbf{U}}\left[\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right]^{-1}$ and moving everything to the left-hand side gives the *adjoint equation*,

$$\left(\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right)^T\mathbf{\Psi} + \left(\frac{\partial y}{\partial\mathbf{U}}\right)^T = 0. \tag{18}$$

Due to the transpose on the unsteady Jacobian matrix, the adjoint system is most easily solved by marching backwards in time, as indicated in Figure 21,

Primal unsteady Jacobian

$$\frac{\partial\mathbf{R}}{\partial\mathbf{U}} = \begin{bmatrix} \star & & & & & & \\ \star & \star & & & & & \\ & \star & \star & & & & \\ & & \star & \star & & & \\ & & & \star & \star & & \\ & & & & \star & \star & \\ & & & & & \star & \star \end{bmatrix}$$

Adjoint unsteady Jacobian

$$\left(\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right)^T = \begin{bmatrix} \star & \star & & & & & \\ & \star & \star & & & & \\ & & \star & \star & & & \\ & & & \star & \star & & \\ & & & & \star & \star & \\ & & & & & \star & \star \\ & & & & & & \star \end{bmatrix}$$

Figure 21: Illustration of the primal and adjoint unsteady residual Jacobians for a backward Euler temporal discretization ($\star$ = block of spatial unknowns). The structure of the primal Jacobian makes forward substitution the natural choice for solving the primal system, whereas the adjoint system is most easily solved using *backward substitution*.

Finally, although we have focused on a parameter sensitivity analysis, we will see in the next section that residual perturbations also arise when discrete solutions are viewed from an enriched space, which motivates the use of adjoint solutions in output error estimation.

5.2.2 Output Error Estimation

Without access to infinite resolution, estimating the true numerical error in an output is practically out of reach for general nonlinear problems. We therefore resign ourselves to estimating the output error between two finite-dimensional spaces: a coarse approximation space (subscript H) on which we calculate the state and output, and a fine space (subscript h) relative to which we estimate the error,

$$\begin{array}{llll} \text{coarse space:} & \rightarrow \underbrace{\mathbf{R}_H(\mathbf{U}_H) = 0}_{N_H \text{ equations}} \rightarrow & \underbrace{\mathbf{U}_H}_{\text{state} \in \mathbb{R}^{N_H}} \quad \rightarrow & \underbrace{y_H(\mathbf{U}_H)}_{\text{output (scalar)}} \\ \text{fine space:} & \rightarrow \underbrace{\mathbf{R}_h(\mathbf{U}_h) = 0}_{N_h \text{ equations}} \rightarrow & \underbrace{\mathbf{U}_h}_{\text{state} \in \mathbb{R}^{N_h}} \quad \rightarrow & \underbrace{y_h(\mathbf{U}_h)}_{\text{output (scalar)}} \end{array}$$

We would like to measure the output error in the coarse solution relative to the fine space,

$$\text{output error:} \;\; \delta y \equiv y_H(\mathbf{U}_H) - y_h(\mathbf{U}_h). \tag{19}$$

The fine space is typically constructed by uniformly refining the discretization or by increasing the approximation order. For example, in the present unsteady studies, we use approximation order increase: $p \to p+1$, $r \to r+1$. We assume that the fine approximation space contains the coarse approximation space, so that the following lossless state injection, $\mathbf{U}_h^H$, is possible, $\mathbf{U}_h^H \equiv \mathbf{I}_h^H \mathbf{U}_H$, where $\mathbf{I}_h^H$ is the coarse-to-fine state injection (prolongation) operator. On the fine space, the exact solution $\mathbf{U}_h \in \mathbb{R}^{N_h}$ would give us zero fine-space residuals, but the state injected from the coarse space will generally not be the fine space solution and hence will not give us zero fine-space residuals, $\mathbf{R}_h(\mathbf{U}_h^H) \neq \mathbf{0}$. Instead, the injected coarse state solves a *perturbed* fine-space problem, in which the left-hand-side residual perturbation is $\delta\mathbf{R}_h = -\mathbf{R}_h(\mathbf{U}_h^H) \neq \mathbf{0}$. The fine-space adjoint, $\mathbf{\Psi}_h$, tells us to expect an output perturbation given by the inner product between the adjoint and the residual perturbation,

$$\underbrace{y_h(\mathbf{U}_h^H) - y_h(\mathbf{U}_h)}_{\approx \delta y} = \mathbf{\Psi}_h^T \delta\mathbf{R}_h = -\mathbf{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H). \tag{20}$$

This derivation assumes small perturbations in $\mathbf{U}$ and $\mathbf{R}$ when the output or equations are nonlinear, and that $y_H(\mathbf{U}_H) = y_h(\mathbf{U}_h^H)$, which is true if the output definition does not change between the coarse and fine spaces. In summary, we have

$$\delta y \approx -\mathbf{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H). \tag{21}$$

Note that this error estimate does not require the fine-space primal solution, $\mathbf{U}_h$.

The error estimate in (21) uses the adjoint on the fine space, $\mathbf{\Psi}_h$. Obtaining $\mathbf{\Psi}_h$ requires solving a larger system compared to the primal, but this system is linear. If the size of the adjoint system makes such a solve impractical, approximations such as reconstruction or cheap iterative smoothing are possible.

5.2.3 Space-Time Mesh Adaptation

The output error estimate derived in the previous section provides error bars on quantities of interest from numerical simulations. However, the benefit of these estimates extends beyond the error bars. Because the output error estimate takes the form of a weighted residual, and because local mesh refinement decreases residuals, the error estimate provides a means of targeting for refinement areas of the computational domain that give rise to the output error. This is the key idea of output-based mesh adaptation.

The adjoint-weighted residual error estimate in (21) can be localized to individual space-time elements by simply separating out contributions to δy from each element,

$$\delta y = \sum_{n=1}^{N_\text{slab}} \sum_{e=1}^{N_\text{elem}} \varepsilon_{k,e}, \qquad \varepsilon_{k,e} = -\left[\mathbf{\Psi}_h^T \mathbf{R}_h(\mathbf{U}_h^H)\right]_{k,e} \tag{22}$$

We can then define an error indicator as $\epsilon_{k,e} = \left|\varepsilon_{k,e}\right|$. However, this indicator does not tell us whether the error is from spatial or temporal resolution, which is needed to identify the most efficient adaptation strategy. For example, if the discretization were under-resolved in the time domain, the temporal error would dominate in $\epsilon_{k,e}$, so refining the elements in space would not be desirable. We calculate the error indicator *anisotropy* using separate projections of the fine-space adjoint onto semi-coarsened spatial and temporal spaces [17, 18],

$$\beta^{\text{space}}_{e,k} = \frac{|\varepsilon^{\text{space}}_{e,k}|}{|\varepsilon^{\text{space}}_{e,k}| + |\varepsilon^{\text{time}}_{e,k}|}, \qquad \beta^{\text{time}}_{e,k} = 1 - \beta^{\text{space}}_{e,k}, \tag{23}$$

$$\underbrace{\epsilon_e = \sum_{k=1}^{N_{\text{slab},H}} \epsilon_{k,e}\beta^{\text{space}}_{e,k},}_{\text{total error on element } e} \qquad \underbrace{\epsilon_k = \sum_{e=1}^{N_{\text{elem},H}} \epsilon_{k,e}\beta^{\text{time}}_{e,k}.}_{\text{total error on time slab } k}$$

We then use a greedy fixed degree-of-freedom (DOF) growth adaptation strategy in which spatial and temporal refinement options are chosen based on a figure of merit, which is the error addressed per DOF added. Element and slab bisection are used for refinement. Adaptation starts from a coarse space-time mesh and repeats the unsteady solution on successively finer meshes, as illustrated in Figure 22.

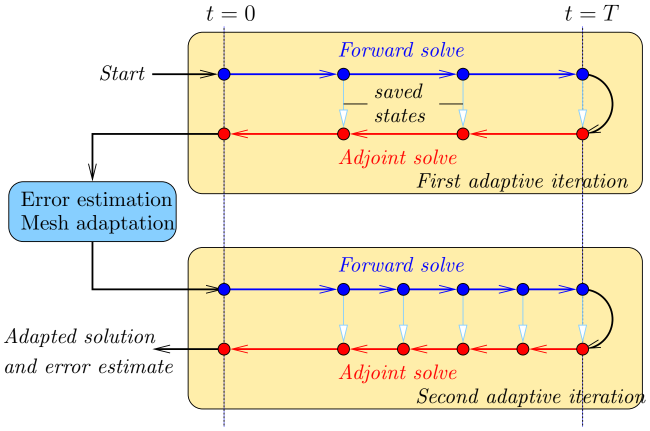


Figure 22: Schematic of output-based adaptation for unsteady problems. Starting from an initial (coarse) space-time mesh, at each adaptive iteration both the primal and the adjoint equations are solved in order to estimate the output error and to generate a new space-time mesh for the next iteration.

5.2.4 Results

We apply our space-time output-based error estimation and adaptation strategy to the one-dimensional radiative shock-tube problem. The starting mesh is quite coarse: 48 spatial elements and 10 time slabs. We compare three adaptive methods:

1. Output-based refinement driven by the adjoint.

2. Refinement driven by spatial and temporal jumps in the state.

3. Uniform refinement of the spatial and temporal mesh.

Figure 23 shows convergence of the output for each method versus the total number of degrees of freedom (space and time) in the mesh. Note that for the output-adapted results, we are able to plot numerical “error bars”, taken here as ± the summed error indicator value. This is a conservative estimate of the output error in that no elements can contribute “negative error”. The output-based result converges most quickly to the exact value, which was computed from a highly-resolved run. The benefit over uniform refinement is over two orders of magnitude in total degrees of freedom.

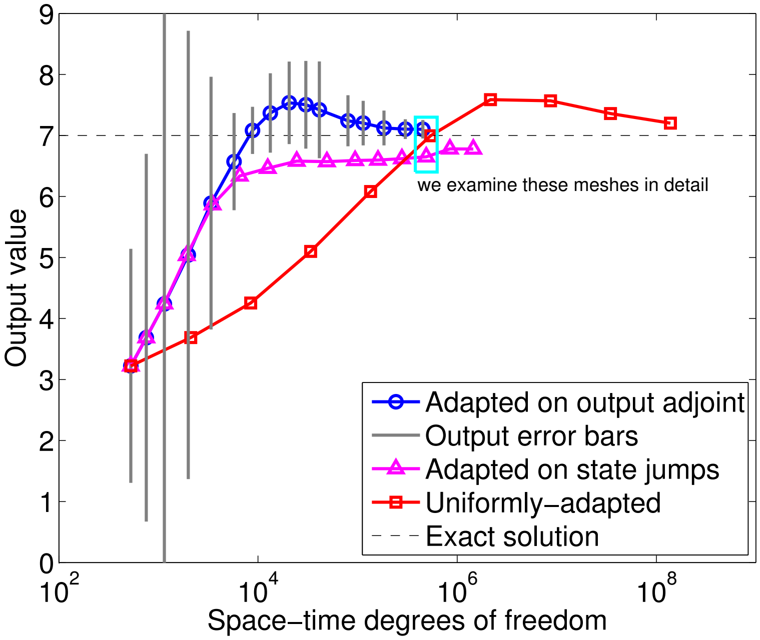


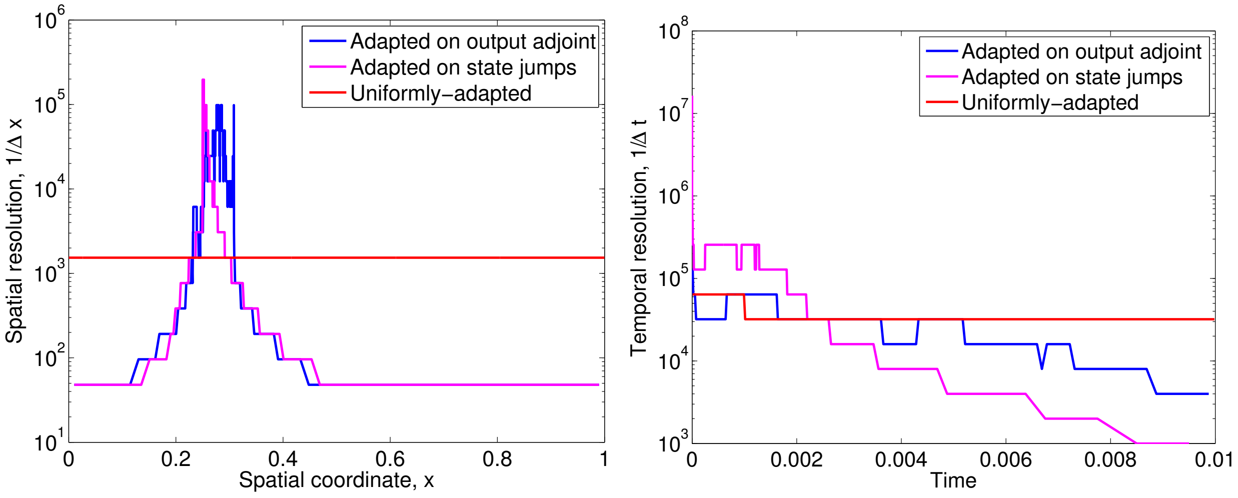
Figure 23: Convergence of a point-density output in an unsteady one-dimensional radiative shock tube problem using three different adaptive indicators. The output-based approach requires the solution of an auxiliary adjoint problem but yields the smallest space-time mesh for accurate output predictions and comes with a built-in error estimate.

For three meshes of similar total degrees of freedom indicated in Figure 23, we plot the spatial and temporal resolutions in Figure 24. Both the output-based and the jump-based indicators target generally similar regions of space, although the output-based refinement is shifted towards the output location, hence providing the observed improved accuracy. In the temporal mesh, output-based refinement also allocates more resolution later in the simulation compared to jump-based refinement.

Extension of the output-based strategy to multiple dimensions or more complex physics is possible in terms of purely mechanistic considerations, as the discrete adjoint equations and error estimation formulas are not specific to one spatial dimension or one physical model. However, for unsteady problems, we do have to proceed cautiously when working with systems that may exhibit chaotic behavior, such as that induced by hydrodynamic instabilities. In such cases, a bounded unsteady adjoint solution may not exist, and the error estimation framework as presented falls apart. Indeed, error estimation the chaotic regime is a subject of ongoing research.

6.0 CONCLUSIONS

The CRASH project completed its primary project goals. These included carrying out and post-processing experiments on “simple” systems in support of uncertainty quantification and prediction, and on the complex system for assessing the predictive capability of the model. Another goal was development of the simulation code, and this too saw rapid progress during the project, so that by late 2011, the CRASH code was routinely



(a) Spatial resolution

(b) Spatial resolution

Figure 24: Comparison of spatial and temporal mesh resolutions (inverse of mesh size) in the adapted meshes identified in Figure 23. All meshes have roughly the same total space-time degrees of freedom but the adaptive methods distribute these degrees of freedom differently over the spatial and temporal domains.

generating run sets in support of predictive model development. An assessment of code convergence and solution verification revealed that for complex simulations, numerical dissipation plays a stabilizing role, and several techniques were explored for estimating the impact of numerical errors. Most importantly, the CRASH project completed an assessment of predictive capability by constructing a Bayesian model for predicting an output of interest in a previously unexplored region of input space, using numerical simulations and experimental data from an explored region of input space. The predictions differed from the actual observations of the complex system by about 10%, which was on the order of the experimental variability and achieved the stated goal.

ACKNOWLEDGMENTS

This manuscript summarizes some of the work of many faculty, research scientists, and students over a five-year long project at the University of Michigan, Texas A&M University, and Simon Fraser University. Due to the large scope of the project, this summary is not comprehensive. However, the results shown represent the culmination of a lot of hard work by dedicated personnel, and I would like to gratefully acknowledge the entire CRASH project team for their efforts. Support for this research was provided by the DOE NNSA/ASC under the Predictive Science Academic Alliance Program by grant number DEFC52-08NA28616.

REFERENCES

[1] Drake, R., Doss, F., McClarren, R., Adams, M., Amato, N., Bingham, D., Chou, C., DiStefano, C., Fidkowski, K., Fryxell, B., Gombosi, T., Grosskopf, M., Holloway, J., van der Holst, B., Huntington, C., Karni, S., Krauland, C., Kuranz, C., Larsen, E., van Leer, B., Mallick, B., Marion, D., Martin, W., Morel, J., Myra, E., Nair, V., Powell, K., Rauchwerger, L., Roe, P., Rutter, E., Sokolov, I., Stout, Q., Torralva, B.,

Toth, G., Thornton, K., and Visco, A., "Radiative effects in radiative shocks in shock tubes," *High Energy Density Physics*, Vol. 7, 2011, pp. 130–140.

[2] Reighard, A. and Drake, R., "The formation of a cooling layer in a partially optically thick shock," *Astrophysics and Space Science*, Vol. 307, 2007, pp. 121–125.

[3] Doss, F., Robey, H., Drake, R., and Kuranz, C., "Wall shocks in high-energy density shock tube experiments," *Physics of Plasmas*, Vol. 16, 2009, pp. 112705.

[4] van der Holst, B., Toth, G., Sokolov, I., Powell, K., Holloway, J., Myra, E., Stout, Q., Adams, M., Morel, J., Karni, S., Fryxell, B., and Drake, R., "CRASH: a block-adaptive-mesh code for radiative shock hydrodynamics – implementation and verification," *The Astrophysics Journal Supplement*, Vol. 194, 2011, pp. 23.

[5] van der Holst, B., Toth, G., Sokolov, I., Daldorff, L., Powell, K., and Drake, R., "Simulating radiative shocks in nozzle shock tubes," *High Energy Density Physics*, Vol. 8, 2012, pp. 161–169.

[6] van der Holst, B., Toth, G., Sokolov, I., Torralva, B., Powell, K., Drake, R., Klapisch, M., Busquet, M., Fryxell, B., and Myra, E., "Simulating radiative shocks with the CRASH laser package," *High Energy Density Physics*, Vol. 9, 2013, pp. 8–16.

[7] Larsen, J. and Lane, S., "Hyades: a plasma hydrodynamics code for dense plasma studies," *Journal of Quantitative Spectroscopy and Radiative Transfer*, Vol. 51, 1994, pp. 179–186.

[8] Chakraborty, A., Mallick, B., McClaren, R., Kuranz, C., Bingham, D., Grosskopf, M., Rutter, E., Stripling, H., and Drake, R., "Spline-Based Emulators for Radiative Shock Experiments With Measurement Error," *Journal of the American Statistical Association*, Vol. 108, No. 502, 2013, pp. 411–428.

[9] Kennedy, M. C. and O'Hagan, A., "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B*, Vol. 63, 2001, pp. 425–464.

[10] Hastings, W., "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, Vol. 57, No. 1, 1970, pp. 97–109.

[11] Goh, J., Bingham, D., Holloway, J., Grosskopf, M., Kuranz, C., and Rutter, E., "Prediction and Computer Model Calibration Using Outputs From Multifidelity Simulators," *High Energy Density Physics*, Vol. 55, No. 4, 2013, pp. 501–512.

[12] Lowrie, R. and Morel, J., "Discontinuous Galerkin for Hyperbolic Systems with Stiff Relaxation," Los Alamos Scientific Laboratory Technical Report LA-UR-99-2517, 1999.

[13] Roache, P. J., "Error Bars for CFD," AIAA Paper 2003-408, 2003.

[14] Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *American Institute of Aeronautics and Astronautics Journal*, Vol. 49, No. 4, 2011, pp. 673–694.

[15] Fidkowski, K. J. and Luo, Y., "Output-based Space-Time Mesh Adaptation for the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 230, 2011, pp. 5753–5773.

[16] Kast, S. M. and Fidkowski, K. J., “Output-based Mesh Adaptation for High Order Navier-Stokes Simulations on Deformable Domains,” *Journal of Computational Physics*, Vol. 252, No. 1, 2013, pp. 468–494.

[17] Fidkowski, K. J., “Output error estimation strategies for discontinuous Galerkin discretizations of unsteady convection-dominated flows,” *International Journal for Numerical Methods in Engineering*, Vol. 88, No. 12, 2011, pp. 1297–1322.

[18] Fidkowski, K. J., “An Output-Based Dynamic Order Refinement Strategy for Unsteady Aerodynamics,” AIAA Paper 2012-77, 2012.