

Functional Uncertainty Constrained by Law and Experiment

Andrew M. Fraser^{‡*}, Stephen A. Andrews[‡]



Abstract—Many physical processes are modeled by unspecified functions. Here, we introduce the *F_UNCLE* project which uses the Python ecosystem of scientific software to develop and explore techniques for estimating such unknown functions and our uncertainty about them. The work provides ideas for quantifying uncertainty about functions given the constraints of both laws governing the function's behavior and experimental data. We present an analysis of pressure as a function of volume for the gases produced by detonating an imaginary explosive, estimating a *best* pressure function and using estimates of *Fisher information* to quantify how well a collection of experiments constrains uncertainty about the function. A need to model particular physical processes has driven our work on the project, and we conclude with a plot from such a process.

Index Terms—python, uncertainty quantification, Bayesian inference, convex optimization, reproducible research, function estimation, equation of state, inverse problems

Introduction

Some tasks require one to quantitatively characterize the accuracy of models of physical material properties which are based on existing theory and experiments. If the accuracy is inadequate, one must then evaluate whether or not proposed experiments or theoretical work will provide the necessary information. Faced with several such tasks, we have chosen to first work on the equation of state (EOS) of the gas produced by detonating an explosive called PBX-9501 because it is relatively simple. In particular Hixson et al. [hixson2000] describe a model form that roughly defines its properties in terms of an unknown one dimensional function (pressure as a function of volume on a special path) and simple constraints. This EOS estimation problem shares the following challenges with many of the other material models that we must analyze:

1. The uncertain object is a *function*. In principal it has an infinite number of degrees of freedom. In order to implement a Bayesian analysis one must define and manipulate probability measures on sets in function space. We do not know how to define a probability measure on sets in function space, and we do not know how to compare the utility of different families of parametric approximations.

* Corresponding author: afrazer@lanl.gov

‡ XCP-8, Los Alamos National Laboratory

Copyright © 2016 Andrew M. Fraser et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. LA-UR-16-23717

2. Understanding the constraints on the unknown function and the connection between it and experimental measurements requires understanding some detailed physics.
3. Simulations of some of the experiments run for more than a few minutes on high performance computers. The job control is unwieldy as are the mechanisms for expressing trial instances of the unknown functions and connecting them to the simulations.

We are organizing our efforts to address those challenges under the title *F_UNCLE* (Functional UNcertainty Constrained by Law and Experiment). We work in two parallel modes as we develop ideas and algorithms. We write code for a surrogate problem that runs in a fraction of a minute on a PC, and we write code for fitting a model to PBX-9501 in a high performance computing environment. Our focus shifts back and forth as we find and resolve problems. As we have progressed, we have found that improving our software practices makes it easier to express ideas, test them on PCs and implement them on the HPCs. In this paper, we introduce the [F_UNCLE] code, the surrogate problem we have developed for the EOS and our analysis of that problem.

We are also using the project to learn and demonstrate *Best Practices for Scientific Computing* (eg, [wilson2014]) and *Reproducible Research* (eg, [fome12009]). The work is designed to be modular, allowing a wide range of experiments and simulations to be used in an analysis. The code is self documenting, with full docstring coverage, and is converted into online documentation using [sphinx]. Each class has a test suite to allow unit testing. Tests are collected and run using [nose]. Each file is also tested using [pylint] with all default checks enabled to ensure it adheres to Python coding standards, including PEP8. Graphics in this paper were generated using [matplotlib] and the code made use of the [numpy] and [scipy] packages. Among the reasons we chose the Python/SciPy ecosystem, the most important are:

Readable

Writing in Python helps us implement the most important point in [wilson2014]: "Write programs for people, not computers."

Versatile

The Python Standard Library lets us easily connect our scripts to other code, eg, submitting HPC jobs and wrapping libraries written in other languages.

Community support

Because of the large number of other users, it is easy to get answers to questions.

Numerical packages

We use a host of modules from Numpy, SciPy and other sources.

Portable

With the Python/SciPy ecosystem, it is easy to write code that runs on our desktops and also runs in our HPC environment.

The task of mapping measurements to estimates of the characteristics of models for the physical processes that generated them is called an *inverse problem*. Classic examples include RADAR, tomography and image estimation. Our problems differ from those in the diverse and indirect nature of the measurements, the absence of translation invariance and in the kinds of constraints. [F_UNCLE] uses constrained optimization and physical models with many degrees of freedom to span a large portion of the allowable function space while strictly enforcing constraints. The analysis determines the function that maximizes the *a posteriori* probability (the MAP estimate) using simulations to match K data-sets. We characterize how each experiment constrains our uncertainty about the function in terms of its *Fisher information*.

As a surrogate problem, we have chosen to investigate the equation of state (EOS) for the products-of-detonation of a hypothetical High Explosive (HE). The EOS relates the pressure to the specific volume of the products-of-detonation mixture. We follow traditional practice (eg, [ficket2000]) and constrain the function to be positive, monotonically decreasing and convex. To date we have incorporated two examples of experiments: the detonation velocity of a *rate stick* of HE, and the velocity of a projectile driven by HE. The behavior of both these experiments depend sensitively on the EOS function.

The following sections describe the choices made in modeling the EOS function, the algorithm used for estimating the function and the use of the Fisher information to characterize the uncertainty about the function. Results so far indicate optimization can find good approximations to the unknown functions and that analysis of Fisher information can quantify how various experiments constrain uncertainty about their different aspects. While these preliminary results are limited to an illustration of the ideas applied to synthetic data and simple models, the approach can be applied to real data and complex simulations. A plot from work on estimating the EOS of the high explosive PBX-9501 appear in the concluding section.

Fisher Information and a Sequence of Quadratic Programs

Our analysis is approximately Bayesian and Gaussian. We suppose that:

1. Experiments provide data $x = [x_0, \dots, x_n]$, where x_k is the data from the k^{th} experiment
2. We have a likelihood function $p_l(x|\theta) = \prod_k p_k(x_k|\theta)$ in which the data from different experiments are conditionally independent given the parameters θ
3. We have a prior on the parameters $p_p(\theta)$

From those assumptions, one can write the *a posteriori* distribution of the parameters as

$$p(\theta|x) = \frac{p_l(x|\theta)p_p(\theta)}{\int p_l(x|\phi)p_p(\phi)d\phi}. \quad (1)$$

Rather than implement Equation (1) exactly, we use a Gaussian approximation calculated at

$$\hat{\theta} \equiv \operatorname{argmax}_{\theta} p(\theta|x). \quad (2)$$

Since θ does not appear in the denominator on the right hand side of Equation (1), in a Taylor series expansion of the log of the *a posteriori* distribution about $\hat{\theta}$ the denominator only contributes a constant added to expansions of the log of the likelihood and the log of the prior, and

$$\begin{aligned} \log(p(\theta|x)) &= \log\left(\frac{p_l(x|\hat{\theta})p_p(\hat{\theta})}{\int p_l(x|\phi)p_p(\phi)d\phi}\right) \\ &+ \frac{1}{2}(\theta - \hat{\theta})^T \left(\frac{d^2 \log(p_l(x|\phi))}{d\phi^2} + \frac{d^2 \log(p_p(\phi))}{d\phi^2} \right)_{\phi=\hat{\theta}} (\theta - \hat{\theta}) \\ &+ R \\ &\equiv C + \frac{1}{2}(\theta - \hat{\theta})^T H (\theta - \hat{\theta}) + R. \end{aligned}$$

Dropping the higher order terms in the remainder R in leaves the normal or Gaussian approximation

$$\begin{aligned} \theta|x &\sim \mathcal{N}(\hat{\theta}, \Sigma = H^{-1}) \\ p(\theta|x) &= \frac{1}{\sqrt{(2\pi)^{\dim|\Sigma}}} \exp\left(-\frac{1}{2}(\theta - \hat{\theta})^T \Sigma^{-1} (\theta - \hat{\theta})\right). \end{aligned}$$

With this approximation, experiments constrain the *a posteriori* distribution by the second derivative of their log likelihoods. Quoting Wikipedia: “If $p(x|\theta)$ is twice differentiable with respect to θ , and under certain regularity conditions, then the Fisher information may also be written as

$$\mathcal{I}(\theta) = -\mathbb{E}_X \left[\frac{\partial^2}{\partial \theta^2} \log p(X;\theta) \middle| \theta \right]. \quad (3)$$

[...] The Cramér–Rao bound states that the inverse of the Fisher information is a lower bound on the variance of any unbiased estimator”

Our simulated measurements have Gaussian likelihood functions in which the unknown function only influences the mean. Thus we calculate the second derivative of the log likelihood as follows:

$$\begin{aligned} L &\equiv -\frac{1}{2}(x - \mu(\theta))^T \Sigma^{-1} (x - \mu(\theta)) + C \\ \frac{\partial L}{\partial \theta} &= (x - \mu(\theta))^T \Sigma^{-1} \frac{\partial \mu}{\partial \theta} \\ \frac{\partial^2}{\partial \theta^2} L &= -\left(\frac{\partial \mu}{\partial \theta}\right)^T \Sigma^{-1} \left(\frac{\partial \mu}{\partial \theta}\right) + (x - \mu(\theta))^T \Sigma^{-1} \frac{\partial^2 \mu}{\partial \theta^2} \end{aligned}$$

and

$$\mathbb{E}_X \frac{\partial^2}{\partial \theta^2} L = -\left(\frac{\partial \mu}{\partial \theta}\right)^T \Sigma^{-1} \left(\frac{\partial \mu}{\partial \theta}\right),$$

because $\Sigma^{-1} \frac{\partial^2 \mu}{\partial \theta^2}$ is independent of X and $\mathbb{E}_X (x - \mu(\theta)) = 0$.

Iterative Optimization

We maximize the *log* of the *a posteriori* probability as the objective function which is equivalent to 2. Dropping terms that do not depend on θ , we write the cost function as follows:

$$\begin{aligned} C(\theta) &\equiv -\log(p(\theta)) - \sum_k \log(p(x_k|\theta)) \\ &\equiv \frac{1}{2}(\theta - \mu)^T \Sigma^{-1} (\theta - \mu) - \sum_k \log(p(x_k|\theta)), \end{aligned}$$

where k is an index over a set of independent experiments. We use the following iterative procedure to find $\hat{\theta}$, the *Maximum A posteriori Probability* (MAP) estimate of the parameters:

1. Set $i = 0$ and $\theta_i[j] = \mu[j]$, where i is the index of the iteration and j is index of the components of θ .
2. Increment i
3. Estimate P_i and q_i defined as

$$q_i^T \equiv \left. \frac{d}{d\theta} C(\theta) \right|_{\theta=\theta_{i-1}}$$

$$P_i \equiv \left. \frac{d^2}{d\theta^2} C(\theta) \right|_{\theta=\theta_{i-1}}$$

Since the experiments are independent, the joint likelihood is the product of the individual likelihoods and the log of the joint likelihood is the sum of the logs of the individual likelihoods, ie,

$$q_i^T \equiv (\theta_{i-1} - \mu)\Sigma^{-1} + \sum_k \frac{d}{d\theta} \log(p(x_k|\theta)) \Big|_{\theta=\theta_{i-1}}$$

$$\equiv (\theta_{i-1} - \mu)\Sigma^{-1} + \sum_k q_{i,k}^T$$

$$P_i \equiv \Sigma^{-1} + \sum_k \frac{d^2}{d\theta^2} \log(p(x_k|\theta)) \Big|_{\theta=\theta_{i-1}}$$

$$\equiv \Sigma^{-1} + \sum_k P_{i,k}$$

where in $P_{i,k}$ and $q_{i,k}$, i is the iteration number and k is the experiment number.

4. Calculate the matrix G_i and the vector h_i to express the appropriate constraints².
5. Calculate $\theta_i = \theta_{i-1} + d$ by solving the quadratic program

$$\text{Minimize } \frac{1}{2} d^T P_i d + q_i^T d$$

$$\text{Subject to } G_i d \preceq h_i$$

where \preceq means that for each component the left hand side is less than or equal to the right hand side.

6. If not converged go back to step 2.

This algorithm differs from modern SQP methods as each QP sub-problem is has no knowledge of the previous iteration. This choice is justified as the algorithm converges in less than 5 outer loop iterations. This unconventional formulation helps accelerate convergence as the algorithm does not need multiple outer loop iterations to obtain a good estimate of the Hessian, as in modern SQP methods.

The assumption that the experiments are statistically independent enables the calculations for each experiment k in to be done independently. In the next few sections, we describe both the data from each experiment and the procedure for calculating $P_{i,k}$ and $q_{i,k}$.

Equation of State

For our surrogate problem, we say that the thing we want to estimate, θ , represents the equation of state (EOS) of a gas. We also say that the state of the gas in experiments always lies on an isentrope³ and consequently the only relevant data is the pressure as a function of specific volume (cm^3/gram) of the gas.

2. For our surrogate problem, we constrain the function at the last knot to be positive and have negative slope. We also constrain the second derivative to be positive at every knot. See the [F_UNCLE] code and documentation for more details.

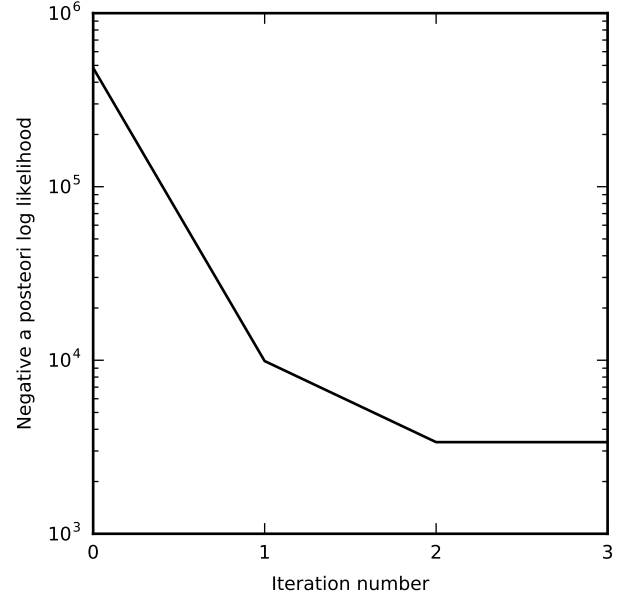


Fig. 1: Convergence history of a typical solution to the MAP optimization problem

For physical plausibility, we constrain the function to have the following properties:

- Positive
- Monotonic
- Convex

Here, let us introduce the following notation:

- v Specific volume
- p Pressure
- f An EOS that maps specific volume to pressure, $f: v \mapsto p$.
- v_0 The minimum relevant volume.
- v_1 The maximum relevant volume.
- \mathcal{F} The set of possible EOS functions, $p(v), v_0 \leq v \leq v_1$

Cubic Splines

While no finite dimensional coordinate scheme can represent every element of \mathcal{F} , the flexibility of cubic splines lets us get close to any element of \mathcal{F} using a finite number of parameters. (An analysis of the efficiency of various representations is beyond the scope of this paper.)

Constraining f to be positive and to be a convex function of v is sufficient to ensure that it is also monotonic. Although we are working on a definition of a probability measure on a sets of functions that obeys those constraints and is further constrained by $\frac{|f(v) - \mu_f(v)|}{\mu_f(v)} \leq \Delta$, for now, we characterize the prior as Gaussian. As we search for the mean of the a posteriori distribution, we enforce the constraints, and the result is definitely not Gaussian. For the remainder of the present work we ignore that inconsistency and use a prior defined in terms of spline coefficients. We start with a nominal EOS

$$\tilde{f}(v) = \frac{F}{v^3}, \text{ where } F \leftrightarrow 2.56 \times 10^9 \text{ Pa at one cm}^3 \text{ g}^{-1} \quad (4)$$

3. In an *isentropic* expansion or compression there is no heat conduction. Our isentropic approximation relies on the expansion being so rapid that there is not enough time for heat conduction.

and over a finite domain we approximate it by a cubic spline with coefficients $\{\tilde{c}_f[j]\}$. Thus c , the vector of spline coefficients, is the set of unknown parameters that we have previously let θ denote. Then we assign a variance to each coefficient:

$$\sigma^2[j] = (c_f[j]\Delta)^2. \quad (5)$$

We set $\Delta = 0.05$. These choices yield:

$$\begin{aligned} \mu_f &\leftrightarrow \{\tilde{c}[j]\} \\ \Sigma_f[i, j] &= \tilde{\sigma}^2[j]\delta_{i,j} \end{aligned}$$

Thus we have the following notation for splines and a prior distribution over \mathcal{F} .

- c_f, b_f Vector of coefficients and cubic spline basis functions that define an EOS. We will use $c_f[j]$ and $b_f[j]$ to denote components.
- μ_f, Σ_f Mean and covariance of prior distribution of EOS. In a context that requires coordinates, we let $\mu_f = (c_f[0], c_f[1], \dots, c_f[n])^T$.

The Nominal and True EOS

For each experiment, data comes from a simulation using a *true* function and each optimization starts from the nominal EOS which is the mean of the prior given in 4. We've made the *true* EOS differ from the nominal EOS by a sum of Gaussian bumps. Each bump is characterized by a center volume v_k , a width w_k and a scale s_k , with:

$$b_k(v) = \frac{s_k F}{v_k^3} e^{-\frac{(v - v_k)^2}{2w_k^2}}$$

Throughout the remainder of this paper, the *true* EOS that we have used to generate pseudo-experimental data is:

$$f(v) = \frac{F}{v^3} + b_0(v) + b_1(v) \quad (6)$$

where: $v_0 = .4 \text{ cm}^3 \text{ g}^{-1}$, $w_0 = .1 \text{ cm}^3 \text{ g}^{-1}$, $s_0 = .25$, $v_1 = .5 \text{ cm}^3 \text{ g}^{-1}$, $w_1 = .1 \text{ cm}^3 \text{ g}^{-1}$, and $s_1 = -.3$.

A Rate Stick

The data from this experiment represent a sequence of times that a detonation shock is measured arriving at locations along a stick of HE that is so thick that the detonation velocity is not reduced by curvature. The code for the pseudo data uses the average density and sensor positions given by Pemberton et al. [pemberton2011] for their *Shot 1*.

Implementation

The only property that influences the ideal measurements of rate stick data is the HE detonation velocity. Code in `F_UNCLE.Experiments.Stick` calculates that velocity following Section 2A of Fickett and Davis [fickett2000] (entitled *The Simplest Theory*). The calculation solves for conditions at what is called the *Chapman Jouguet* (CJ) state. The CJ state is defined implicitly by a line (called the *Rayleigh line*) in the (p, v) plane that goes through (p_0, v_0) , the pressure and volume before detonation, and (p_{CJ}, v_{CJ}) . The essential requirement is that the Rayleigh line be tangent to the isentrope or EOS curve in the (p, v) plane. The slope of the Rayleigh line that satisfies those conditions defines the CJ velocity, V in terms of the following equation:

$$\frac{V^2}{v_0^2} = \frac{p_{CJ} - p_0}{v_0 - v_{CJ}}.$$

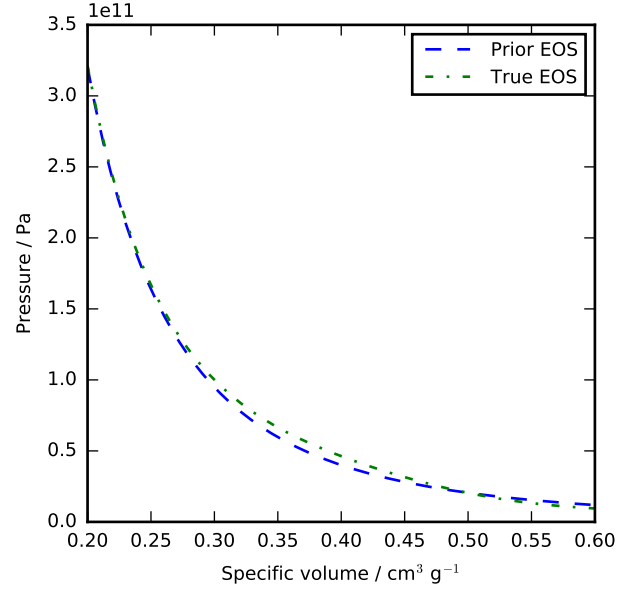


Fig. 2: The prior and nominal true equation of state function. The two models differ most at a specific volume of $0.4 \text{ cm}^3 \text{ g}^{-1}$

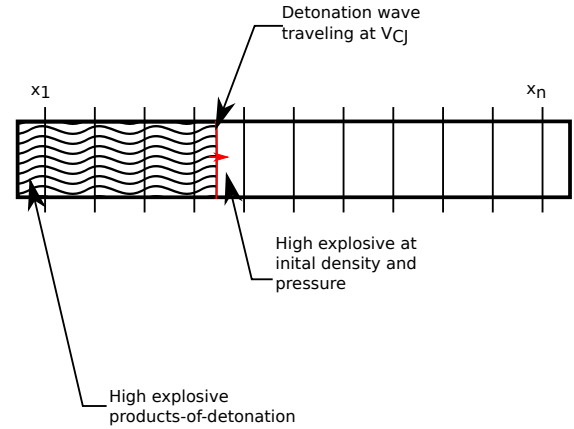


Fig. 3: The rate stick experiment showing the detonation wave propagating through the rate stick at the CJ velocity. Detonation velocity is measured by the arrival time of the shock at the sensors placed along the stick.

For each trial EOS, the `F_UNCLE` code uses the `scipy.optimize.brentq` method in a nested loop to solve for (p_{CJ}, v_{CJ}) . Figure 4 shows the EOS and both the Rayleigh line and the CJ point that the procedure yields.

Comparison to Pseudo Experimental Data

The previous section explained how to calculate the detonation velocity, $V_{CJ}(f)$, but the *experimental* data are a series of times when the shock reached specified positions on the rate-stick. The simulated detonation velocity is related to these arrival times by:

$$t[i] = \frac{x[i]}{V_{CJ}(f)}.$$

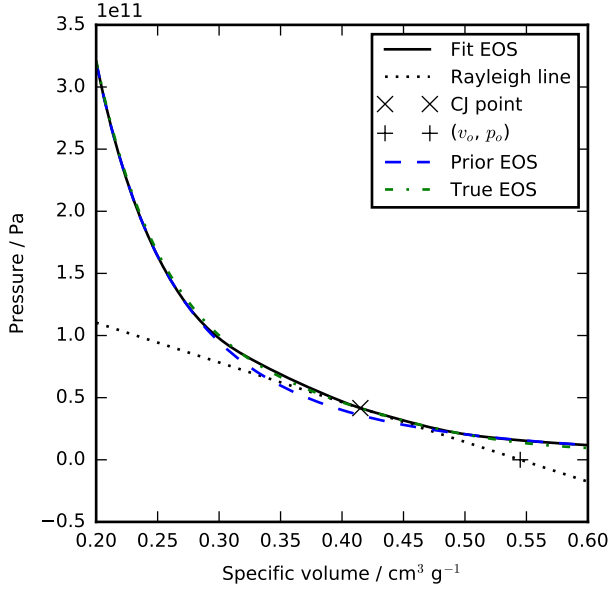


Fig. 4: Isentropes, a Rayleigh line and the CJ conditions. Starting from the isentrope labeled Prior EOS and using data from simulated experiments based on the isentrope labeled True EOS, the optimization algorithm described in the Algorithm section produced the estimate labeled Fit EOS. Solving for the CJ state of Fit EOS isentropes yields a Rayleigh line. The data constrains the isentrope only at v_{CJ} .

where $x[i]$ are the locations of each sensor measuring arrival time.

We let D denote the sensitivity of the set of simulated arrival times to the spline coefficients governing the equation of state, and write:

$$D[i, j] \equiv \frac{\partial t[i]}{\partial c[j]}.$$

We use finite differences to estimate D .

The Gun

The data from this experiment are a time series of measurements of a projectile's velocity as it accelerates along a gun barrel driven by the expanding products-of-detonation of HE. Newton's equation

$$F = ma$$

determines the velocity time series. The product of the pressure from the EOS and the area of the barrel cross section is the force.

Implementation

The position and velocity history of the projectile is generated by the `scipy.integrate.odeint` algorithm. This method solves the differential equation for the projectile position and velocity as it is accelerated along the barrel.

$$\frac{dx(t)}{dt} = v(t) \quad (7)$$

$$\frac{dv(t)}{dt} = \frac{A}{m_{proj}} f\left(\frac{x(t)A}{m_{HE}}\right) \quad (8)$$

where:

- t is time from detonation (assuming the HE burns instantaneously)

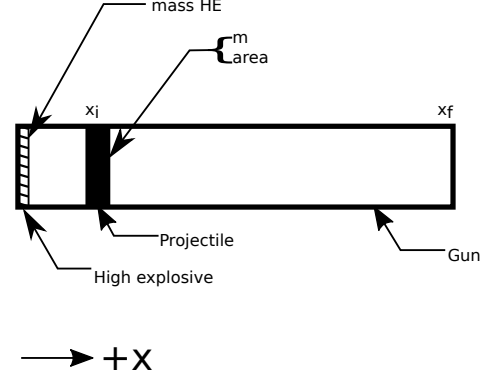


Fig. 5: The gun experiment. The projectile of a given mass and cross-sectional area is accelerated along the barrel by the expanding products of combustion from the high explosives in the barrel.

- $x(t)$ is the position of the projectile along the barrel
- $v(t)$ is the velocity of the projectile
- A is the cross-sectional area of the barrel
- m_{HE} is the mass of high explosives
- m_{proj} is the mass of the projectile
- f is the equation of state which relates the pressure to the specific volume of the HE products-of-detonation

The acceleration is computed based the projectile's mass and the force resulting from the uniform pressure acting on the projectile. This pressure is related to the projectile's position by the EOS, assuming that the projectile perfectly seals the barrel so the mass of products-of-detonation behind the projectile remains constant.

Comparison to Pseudo Experimental Data

We generated *experimental data* using our simulation code with the nominal *true* EOS described previously. These experimental data were a series of times and corresponding velocities. To compare the experiments to simulations, which may use a different time discretization, the simulated response was represented by a spline, and was compared to the experiments at each experimental time stamp.

$$D[i, j] = \frac{\partial \hat{v}(t_{exp}[i])}{\partial c_f[j]} \quad (9)$$

where:

- \hat{v} is the velocity given from the spline fit to simulated $v(t)$ data
- t_{exp} is the times where experimental data were available

Numerical Results

The algorithm was applied to the sets of simulation results and pseudo experimental data for both the rate-stick and gun models. Figure 6 shows the improved agreement between the simulated and *experimental* arrival times for the rate-stick after the algorithm adjusts the equation of state. Similar results are shown in Figure 7 for the gun experiment, where the significant error in velocity history at early times is reduced by an order of magnitude with the optimized EOS model.

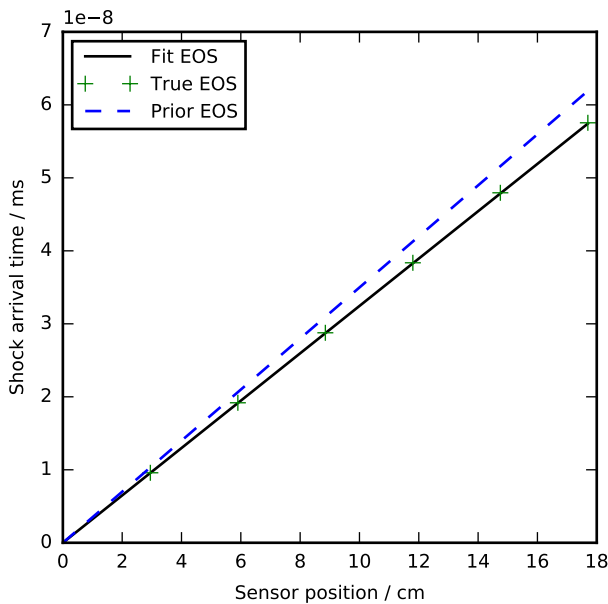


Fig. 6: Fitting an isentrope to rate stick data. Green +’s denote measured shock arrival time at various positions. The blue line represents the shock velocity calculated from the prior EOS, and the black line is the result of the optimization algorithm described in the text.

Fisher Information Matrix

The Fisher information matrix characterizes how tightly the experimental data constrain the spline coefficients. This matrix can be better understood through a spectral decomposition to show the magnitude of the eigenvalues and the eigenvector behavior.

Figure 8 illustrates the spectral decomposition of the Fisher information matrix for the rate-stick experiment. To machine precision, there is only one nonzero eigenvalue. We expect that because only the CJ point on the EOS influences the forecast data, $\mu(c)$. The eigenvector corresponding to this eigenvalue is most influential about the specific volume corresponding to the CJ state.

The Fisher information matrix of the gun experiment is more complex as changes to the EOS affect the entire time history of the projectile velocity. In Figure 9 There is no clear *dominating* eigenvalue, the largest eigenvalue corresponds to an eigenvector which is more influential at smaller projectile displacements while the next three largest eigenvalues correspond to eigenvectors which are more influential across the range of displacements.

These preliminary investigations of the Fisher information matrix show how this matrix can be informative in describing the uncertainty associated with the optimal EOS function determined by the [F_UNCLE] algorithm. Notice that the eigenvectors of the matrix describe functions that are zero for states not visited by the gun experiment.

Conclusion, Caveats and Future Work

We have described an iterative procedure for estimating functions based on experimental data in a manner that enforces chosen constraints. The [F_UNCLE] code implements the procedure, and we used it to make the figures in the previous sections. The code runs on a modest desktop computer and makes the figures in a

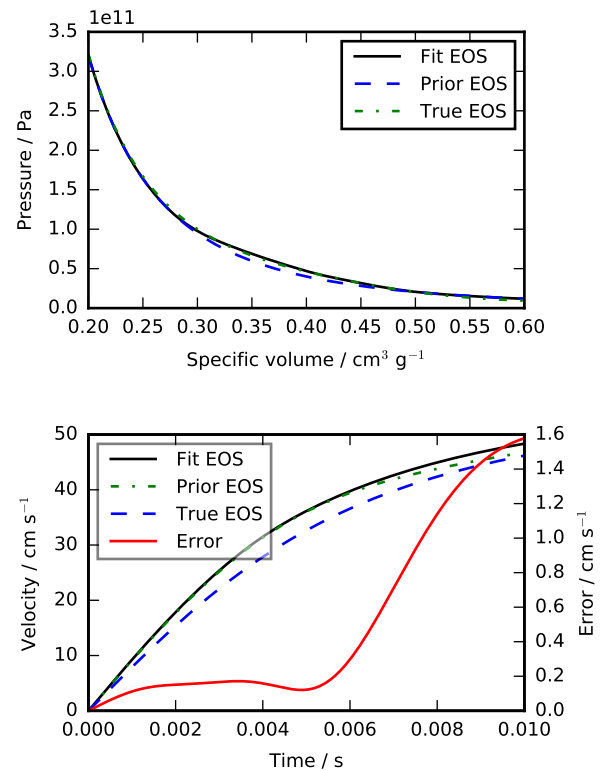


Fig. 7: Estimation of the maximum a posteriori probability parameters of the gun experiment. The True EOS appears in the upper plot, and the optimization starts with the Prior EOS and ends with Fit EOS. The corresponding velocity for the gun as a function of position appears in the lower plot. The estimation also used experimental data from the rate stick.

fraction of a minute. That speed and simplicity allows one to easily try out new ideas and code. We have relied on the [F_UNCLE] code to guide work with real experimental data and simulations on high performance computers that use proprietary software. Figure 10 is the result of applying the ideas presented here to the physical experiments described in [pemberton2011].

The [F_UNCLE] code has been useful for us, and while we believe it could be useful for others, we emphasize that it is a work in progress. In particular:

- The prior is inconsistent. We hope to analyze and perhaps mitigate the effects of that inconsistency in future work.
- The choice of splines is not justified. We plan to compare the performance of coordinate system options in terms of quantities such as bias and variance in future work.
- The optimization procedure is ad hoc and we have not considered convergence or stability. We have already begun to consider other optimization algorithms.

We have designed the [F_UNCLE] code so that one can easily use it to model any process where there is a simulation which depends on a model with an unknown functional form. The self documenting capabilities of the code and the test suites included with the source code will help others integrate other existing models and simulations into this framework to allow it to be applied to many other physical problems.

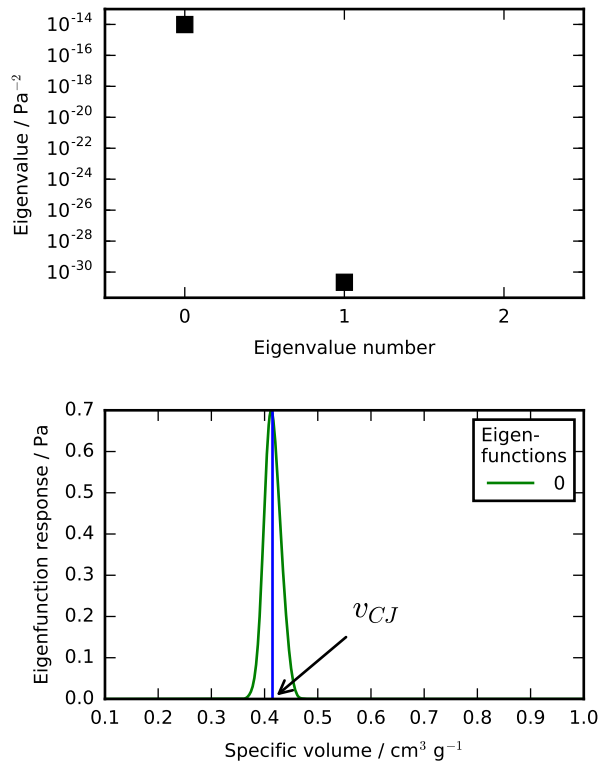


Fig. 8: Fisher Information of the Rate Stick Experiment. The largest two eigenvalues of $\mathcal{I}(\hat{\epsilon})$ appear in the upper plot, and the eigenfunction corresponding to the largest eigenvalue appears in the lower plot.

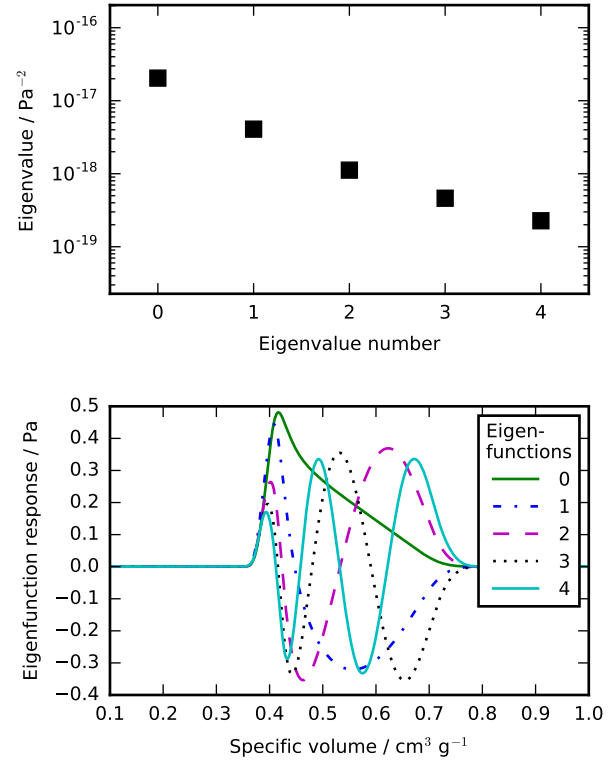


Fig. 9: Fisher Information of the Gun Experiment. The largest five eigenvalues of $\mathcal{I}(\hat{\epsilon})$ appear in the upper plot and the eigenfunctions corresponding to the largest five eigenvalues appear in the lower plot.

REFERENCES

- [cvxopt] Andersen, M. and Vandenberghe, L.. "cvxopt: Convex Optimization Package" <http://cvxopt.org/> [Online; accessed 2016-05-27].
- [ficket2000] Fickett, W. and Davis, W. C., 2000. "Detonation". University of California Press: Berkeley, CA.
- [fomel2009] Fomel, Sergey, and Jon F. Claerbout. "Reproducible research." Computing in Science & Engineering 11.1 (2009): 5-7.
- [F_UNCLE] "F_UNCLE: Functional Uncertainty Constrained by Law and Experiment" https://github.com/fraserphysics/F_UNCLE [Online; accessed 2016-05-27].
- [hill1997] Hill, L. G., 1997. "Detonation Product Equation-of-State Directly From the Cylinder Test". Proc. 21st Int. Symp. on Shock Waves, Great Keppel Island, Australia.
- [hixson2000] Hixson, R. S. et al., 2000. "Release isentropes of overdriven plastic-bonded explosive PBX-9501." *J. Applied Physics* **88** (11) pp. 6287-6293
- [matplotlib] Hunter, J. D.. "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, **9**, 90-95 (2007), DOI:10.1109/MCSE.2007.55
- [nose] "nose: Nose Extends Unittest to Make Testing Easier" <https://pypi.python.org/pypi/nose/1.3.7> [Online; accessed 2016-05-27].
- [numpy] van der Walt, S. , Colbert, C. S. and Varoquaux, G.. "The NumPy Array: A Structure for Efficient Numerical Computation", Computing in Science & Engineering, **13**, 22-30 (2011), DOI:10.1109/MCSE.2011.37
- [pemberton2011] Pemberton et al. "Test Report for Equation of State Measurements of PBX-9501". LA-UR-11-04999, Los Alamos National Laboratory, Los Alamos, NM.
- [pylint] "pylint: Python Code Static Checker" <https://www.pylint.org/> [Online; accessed 2016-05-27].

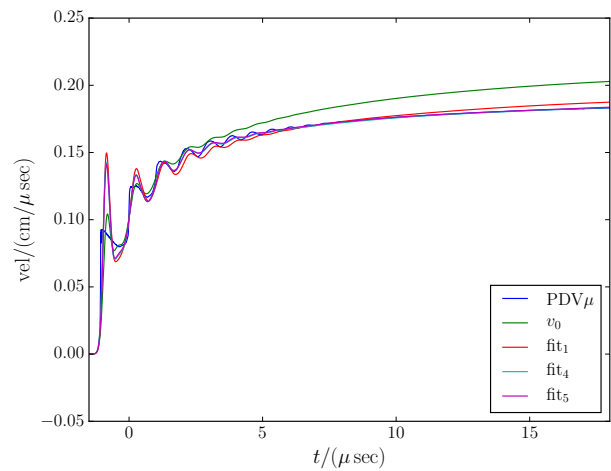


Fig. 10: Improvement of match between true experiments on PBX-9501 and simulations on a high performance computer. The mean of the experimental data is labeled μ , and the optimization scheme yields the EOSs that produce the traces labeled fit_n .

- [scipy] Jones, E., Oliphant, E., Peterson, P., et al. "SciPy: Open Source Scientific Tools for Python", 2001-, <http://www.scipy.org/> [Online; accessed 2016-05-27].
- [sphinx] "sphinx: Python Documentation Generator" <http://www.sphinx-doc.org/> [Online; accessed 2016-05-27].
- [wilson2014] Wilson, Greg, et al. "Best practices for scientific computing." PLoS Biol 12.1 (2014): e1001745.