Code Validation as a Reliability Problem

Tim Trucano

September, 1999

Optimization and Uncertainty Estimation Department

Sandia National Laboratories

Albuquerque, NM 87185

Phone: 844-8812, FAX: 844-0918
Email: tgtruca@sandia.gov

http://sherpa.sandia.gov/9231home/compphys-frame.html

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Consider the following quote:

"Confidence in the [code] modeling approach has reached the point that during the recent Advanced Unitary Penetrator (AUP) Warhead Program at China Lake, the Ordnance Hazard Evaluation Board (OHEB) ruled that actual SD [*sympathetic detonation*] testing was not required and the *results of the SD simulations were adequate for limited fleet introduction of these warheads*. The SD simulations were completed on a part time basis in less than *four weeks* compared to the actual test schedule, which would have been in excess of *one year*." [Emphasis mine]

A Bayesian definition of code Validation?

The process of increasing our subjective degree of belief in the results of a single calculation, or of a set of calculations.

- **"Subjective" does not mean "arbitrary"**
- **"Belief" does not equal "imagination"**

Sandia National Laboratories

Compare this to how the ASCI V&V program actually defined code Validation.

The process of determining the degree to which a computer calculation is an accurate representation of the real world.

- **The V&V "vision" was stated as follows:**

> ***"Establish confidence in the simulations supporting the Stockpile Stewardship Program through systematic demonstration and documentation of the predictive capability of the codes and their underlying models."***

"Predictive content" of a code might be measured by:

Certainly, my "belief" in the calculation increases as the following quantity becomes "small(er)":

$$\boxed{\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3}$$

$$\mathbf{e}_1 = \langle \mathbf{y}_{\text{nature}} - \mathbf{y}_{\text{experiment}} \rangle$$

$$\mathbf{e}_2 = \langle \mathbf{y}_{\text{experiment}} - \mathbf{y}_{\text{calc}} \rangle$$

$$\mathbf{e}_3 = \langle \mathbf{y}_{\text{calc}} - \mathbf{y}_{\text{exact}} \rangle$$

Y_{nature} is "reality";

$y_{experiment}$ is what we "measure";

y_{exact} is the exact solution of the model;

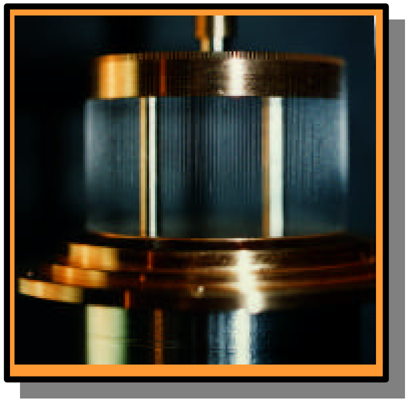
y_{calc} is the code solution of the model;

Confirming that e_1 is small is fundamental.

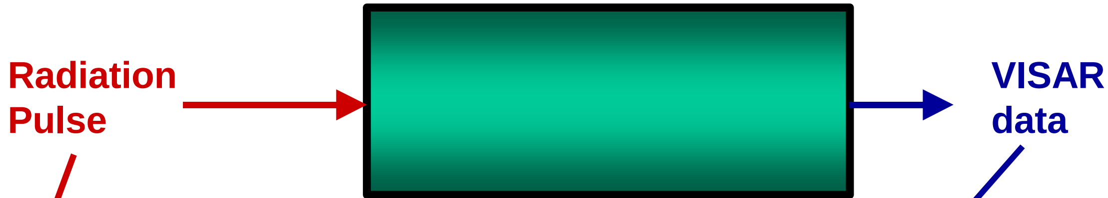
Confirming that e_3 is small is a ***verification*** problem.

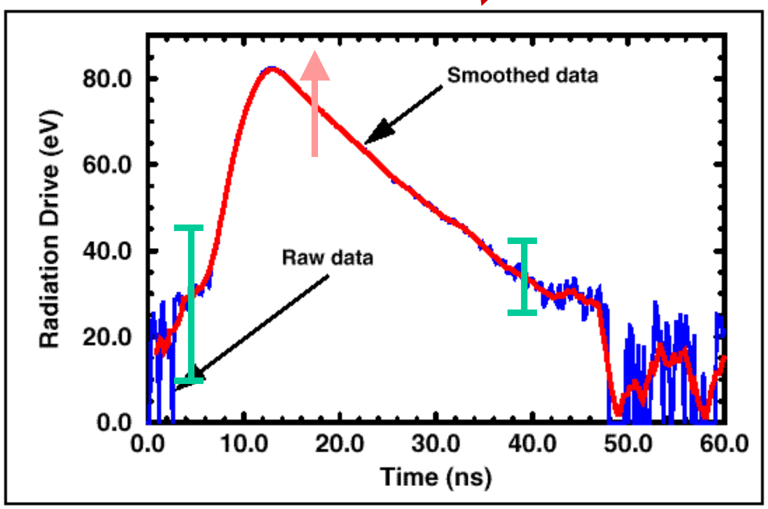
Confirming that e_2 is small is a ***validation*** problem.

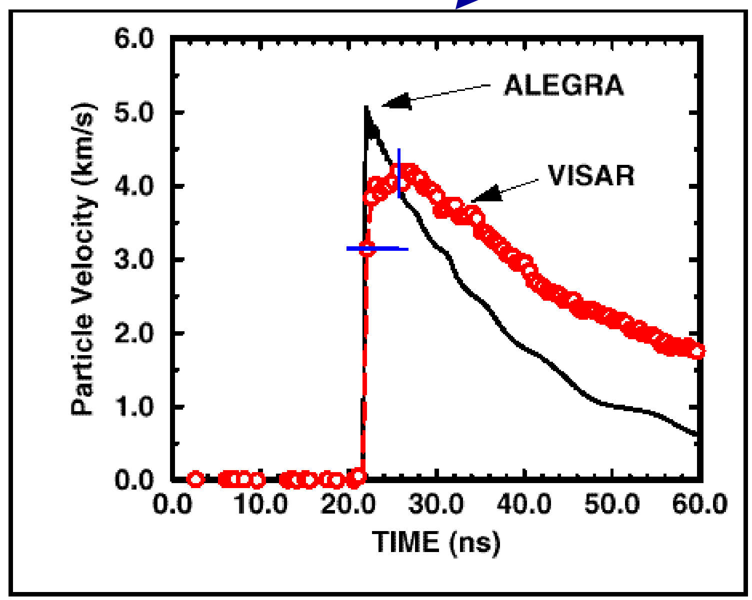
An example of practical uncertainty: Radiation-driven shock waves on the Sandia Z-Machine



Imploding wire array generates radiation







Sandia National Laboratories

The obvious conclusion is -

Uncertainty quantification is important for validation.

We seek to validate multi-physics codes.

Multi-physics often implies **general purpose.**

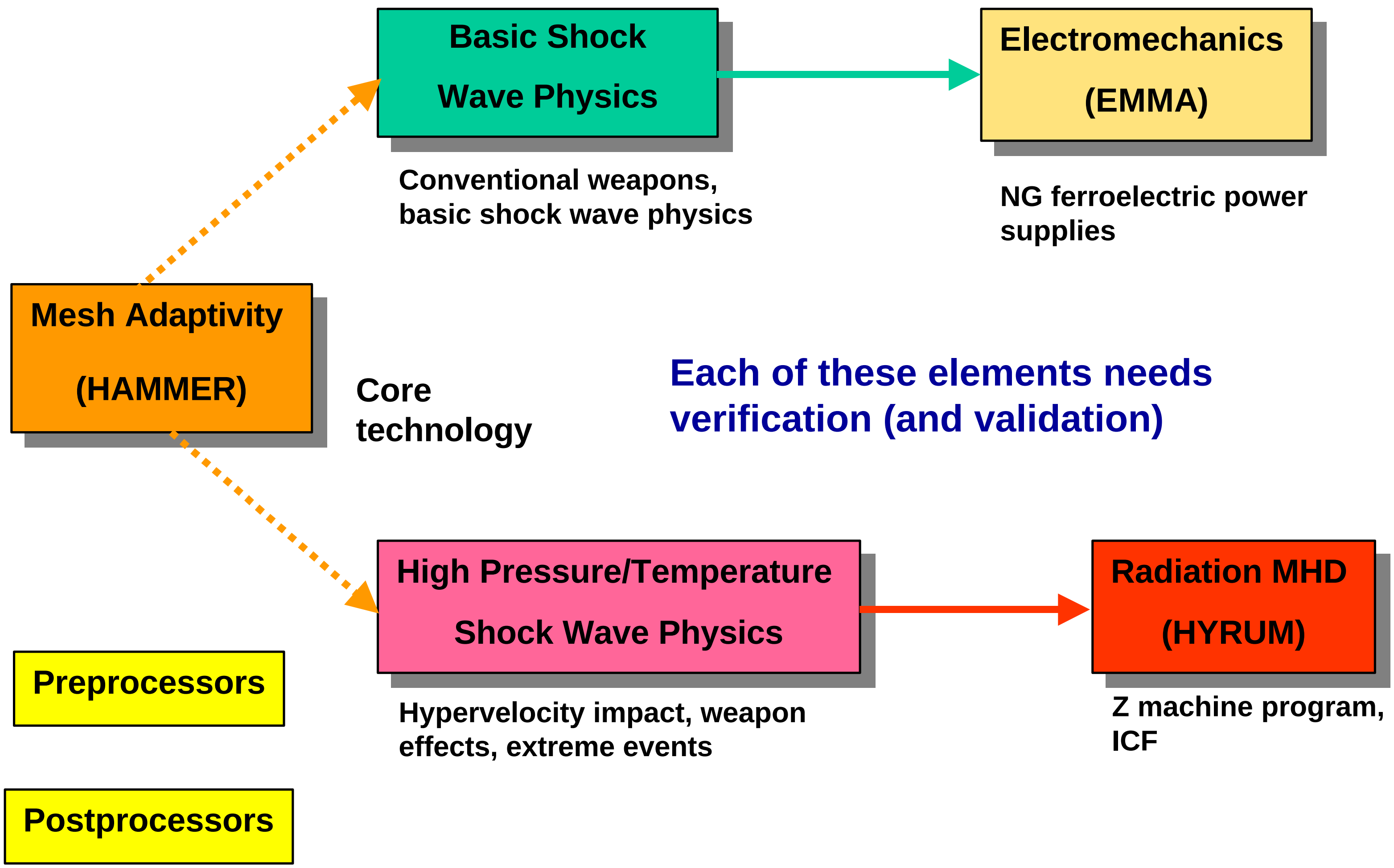
General purpose implies that **users** are a huge component in "correct" application of the code. Code validation does not address user error per se, but we still need to worry about it.

Multi-physics also often implies **research models** are present.

An example is the Sandia code ALEGRA:

- **3-D multi-material Arbitrary Lagrangian Eulerian Radiation-MHD shock wave physics code**
- **C++ (150000 lines) + MPI + C + Fortran, etc. (1,000,000 total including special libraries)**
- **Under development**

Example: ALEGRA is a multi-physics code.



Each of these elements needs verification (and validation)

Sandia National Laboratories

What is "<u>uncertainty quantification</u>" and why do I care?

Uncertainty quantification: an anti-reductionist measure of "error".

The <u>forward prediction</u> problem:

- **Characterize the "input" uncertainty (stochastic, fuzzy, etc)**
- **Propagate this uncertainty through the code**
- **Characterize the resulting output uncertainty**
- ***Refine this characterization via comparison with data***
- **Develop "code reliability" metrics and statements (need requirements)**
- **(Most of this is not rocket science for an initial implementation.)**

Now follow it with <u>backward prediction</u>:

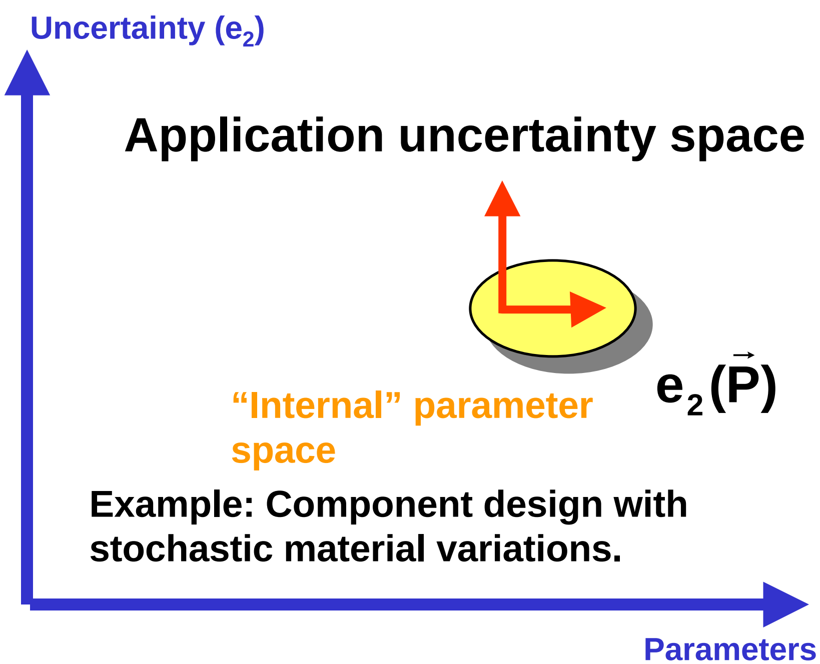
- **Reduce the code uncertainty via the output uncertainty characterization (Bayesian?).**
- **(This IS rocket science.)**

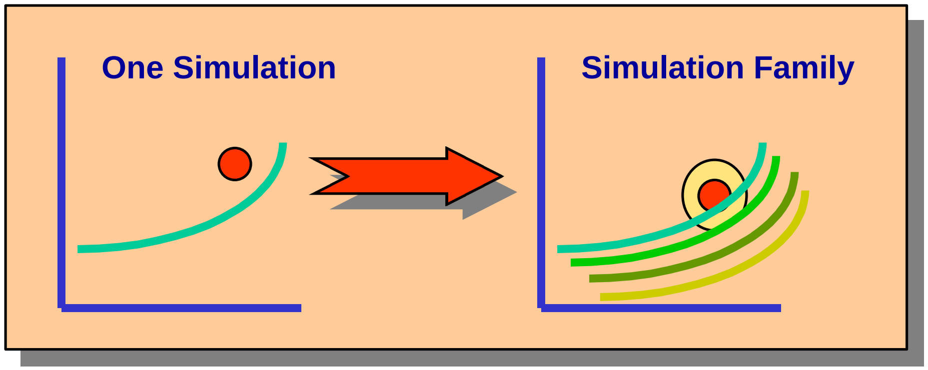
Now <u>optimize</u>:

- **Perform forward/backward prediction sweeps to increase "code reliability" and guide new experiments.**

Sandia National Laboratories

Local validation begins with local uncertainty quantification, a high dimensional problem.





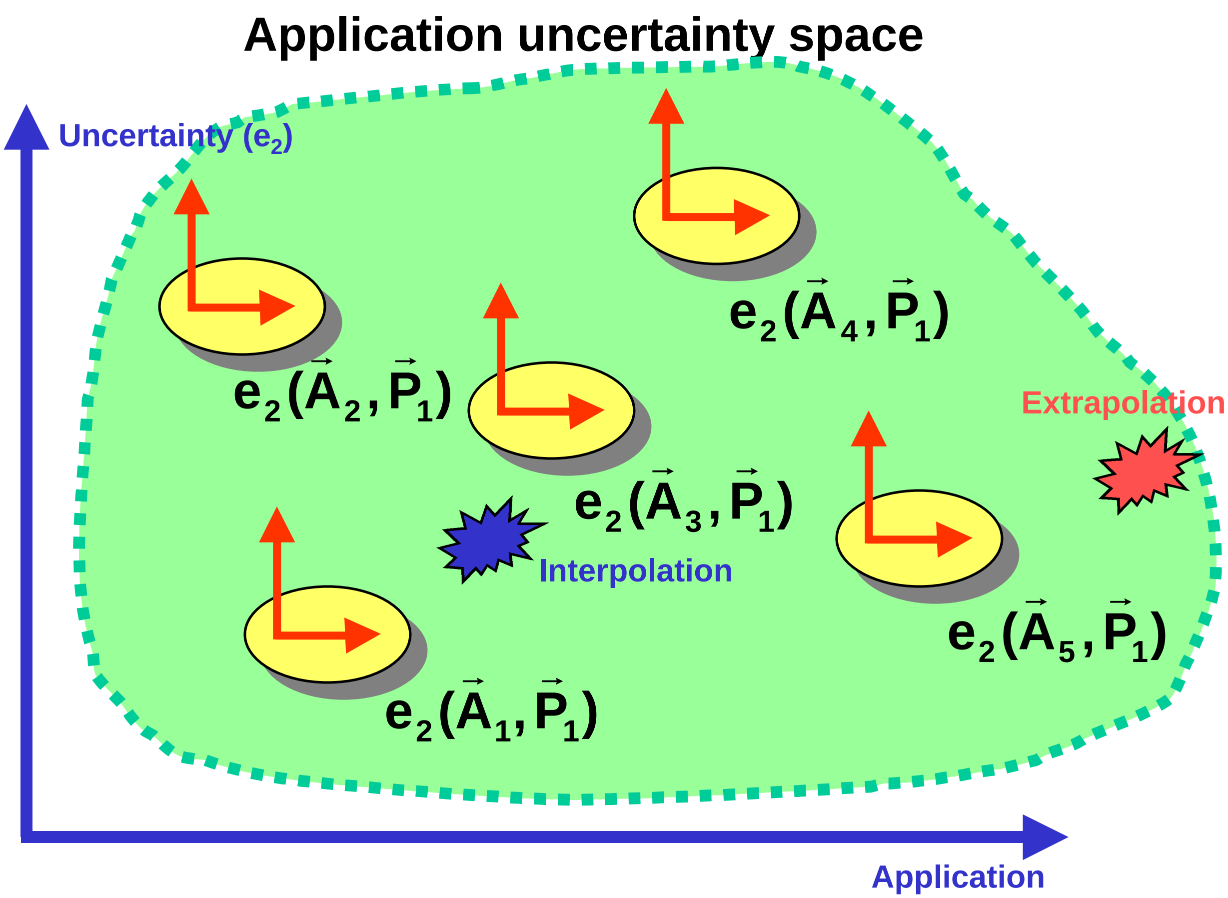
Local uncertainty quantification performs systematic studies of code uncertainty from stochastic treatments of parameter uncertainty. e_2 is a random variable, P is a random vector.

Notice that how uncertainty U is defined is an issue.

Alternatively, think of results being replaced by probability distributions of results. Then, do response surface methods, or other things, to characterize the result family stochastically. The most important issue is this characterization.

The number of parameters can be enormous - *parsimony* is hoped for (but unlikely?).

“Global” validation leads to global uncertainty quantification, an even higher dimensional problem with “internal” and “applications” parameters.



Example: NIF ignition capsule design confidence levels based on uncertainty quantified NOVA design calculation experience.

Uncertainty has two components:

“Local” - uncertainty quantification and local data comparisons.

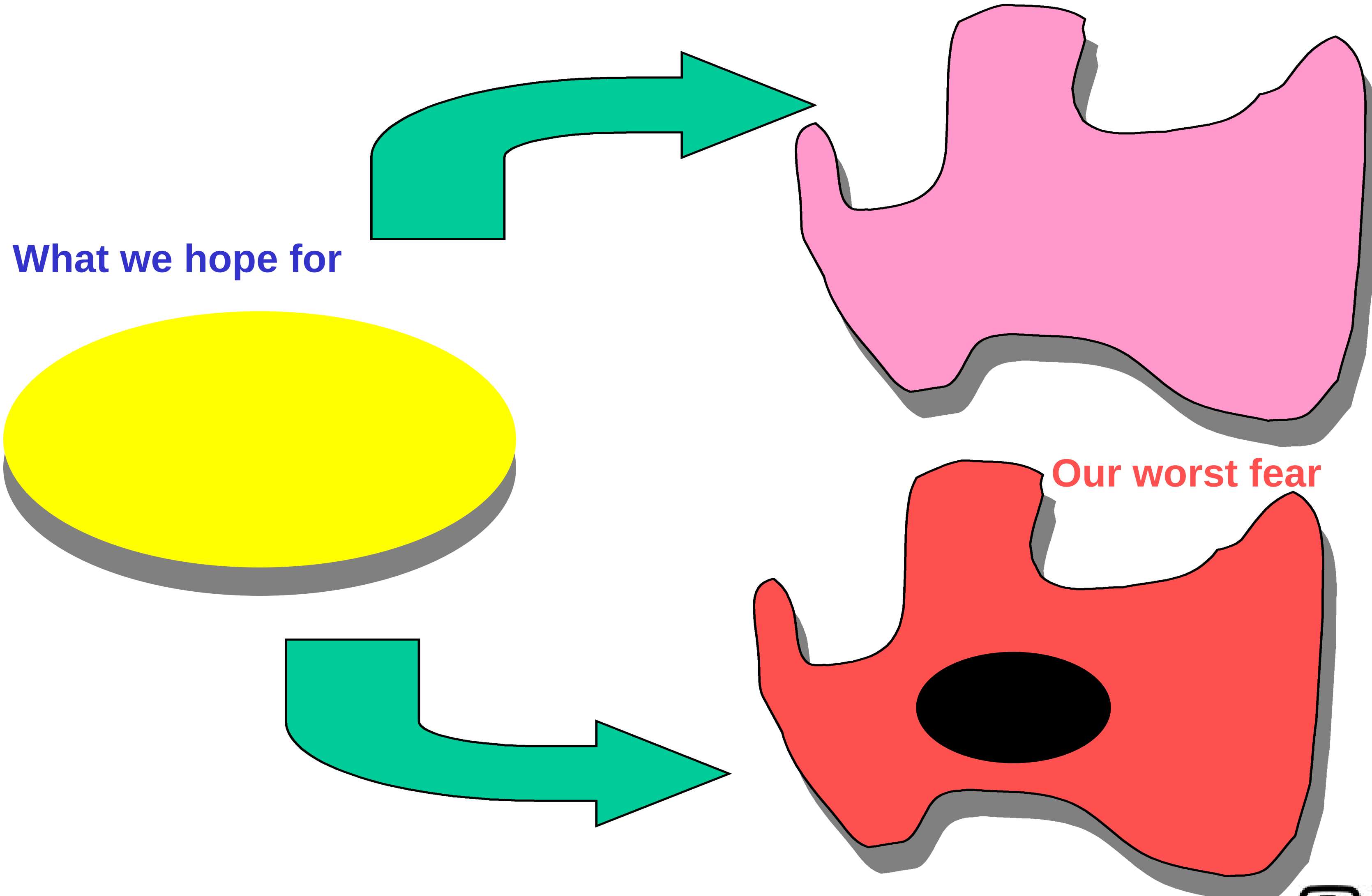
“Global” - system scale uncertainty quantification and global data comparisons.

Can we use spatial statistics methods (like kriging) to characterize e_2(A,P)?

Note that we have simplified the problem by assuming that the internal parameter functional dependence is constant over application space. Is this true?

Sandia National Laboratories

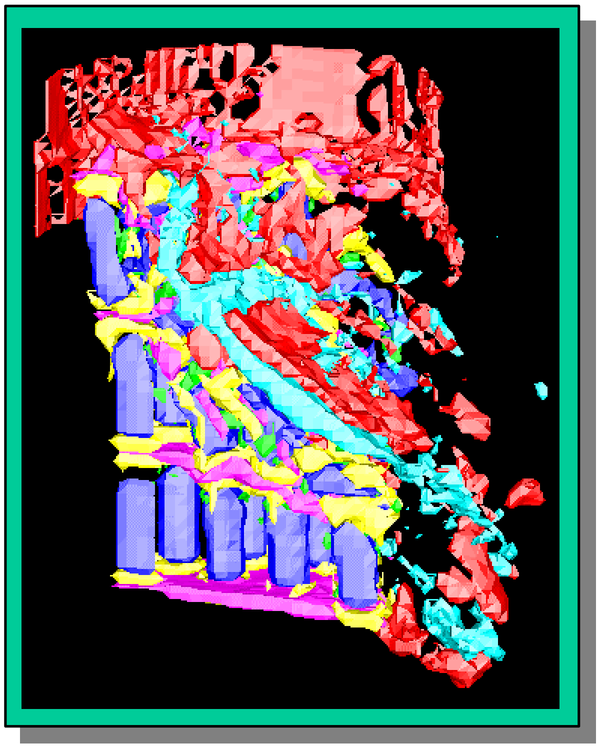
“Uncertainty” probably looks a lot more complex than suggested by the previous figures.

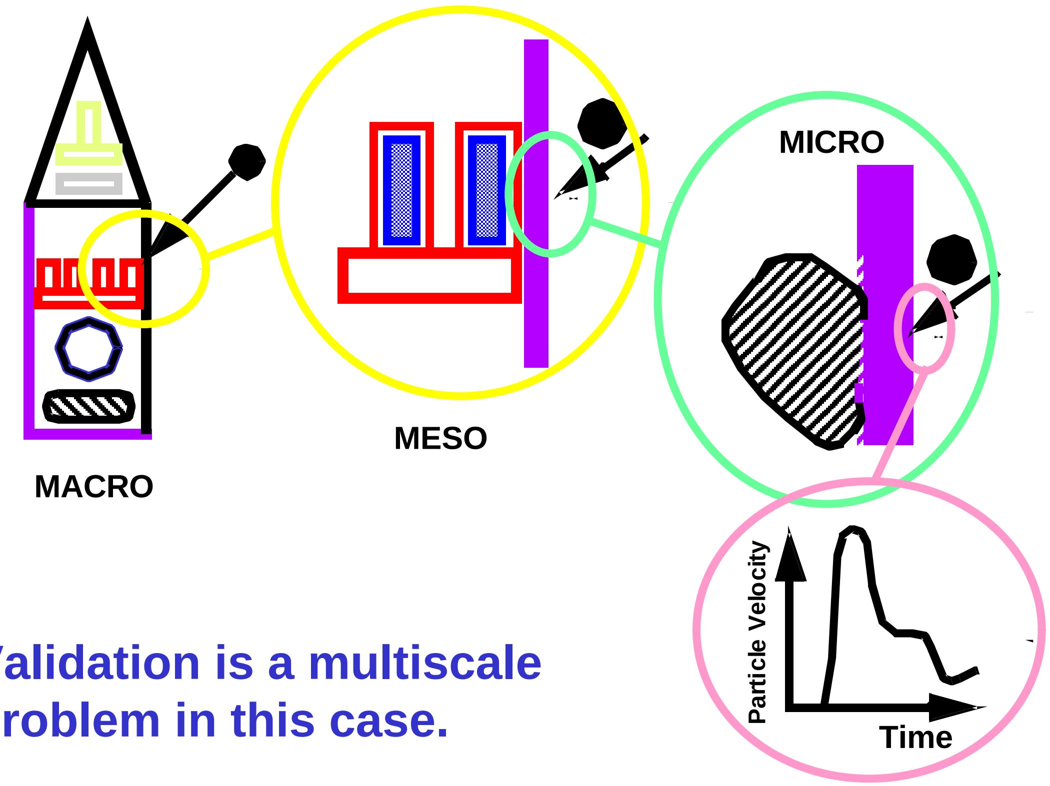


Example - Hypervelocity Impact

Missile defense simulations provide an excellent application for studying predictive complexity.

There are at least six stochastic parameters to begin with: the hit point and the engagement velocity



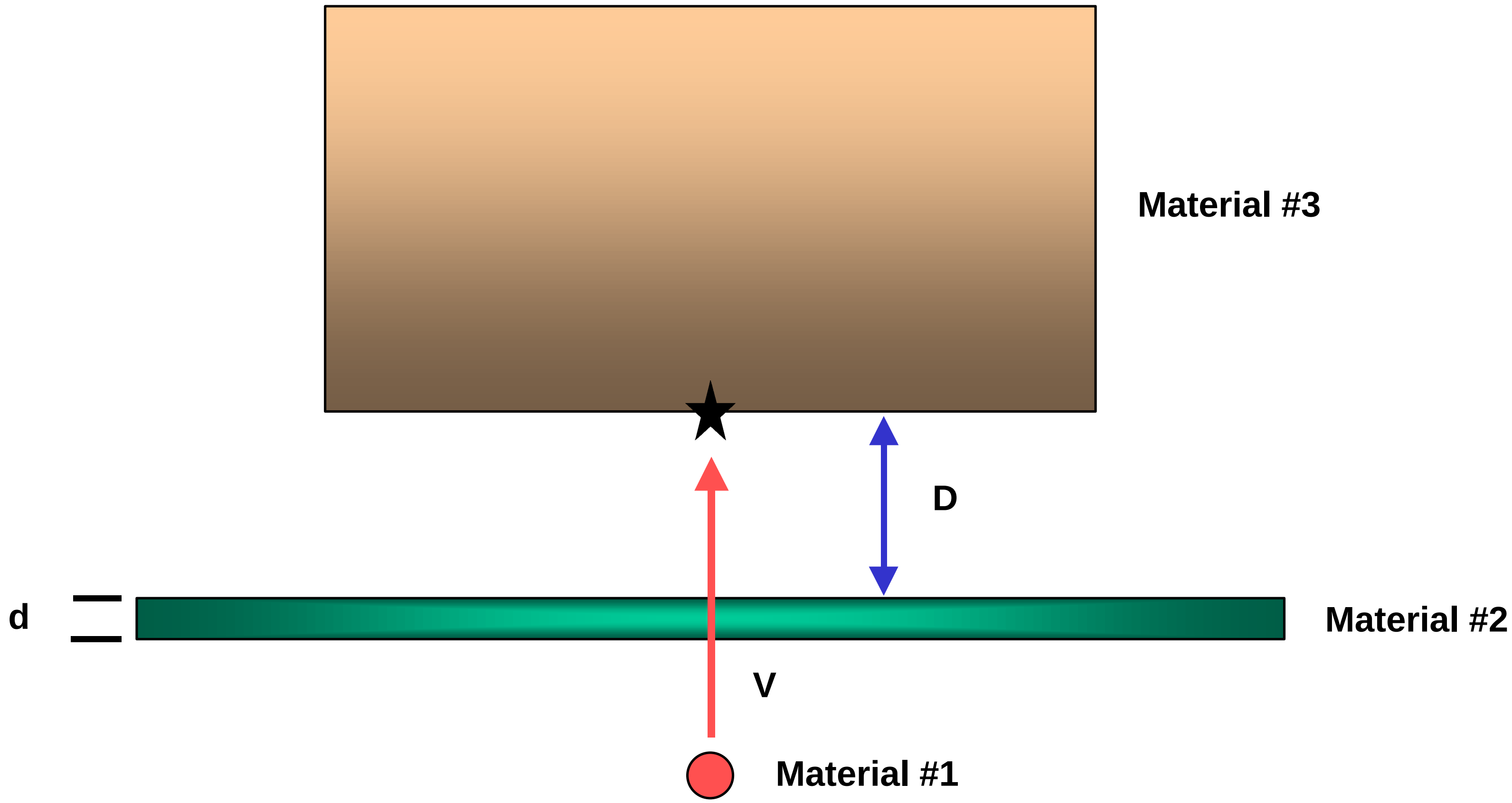


Validation is a multiscale problem in this case.

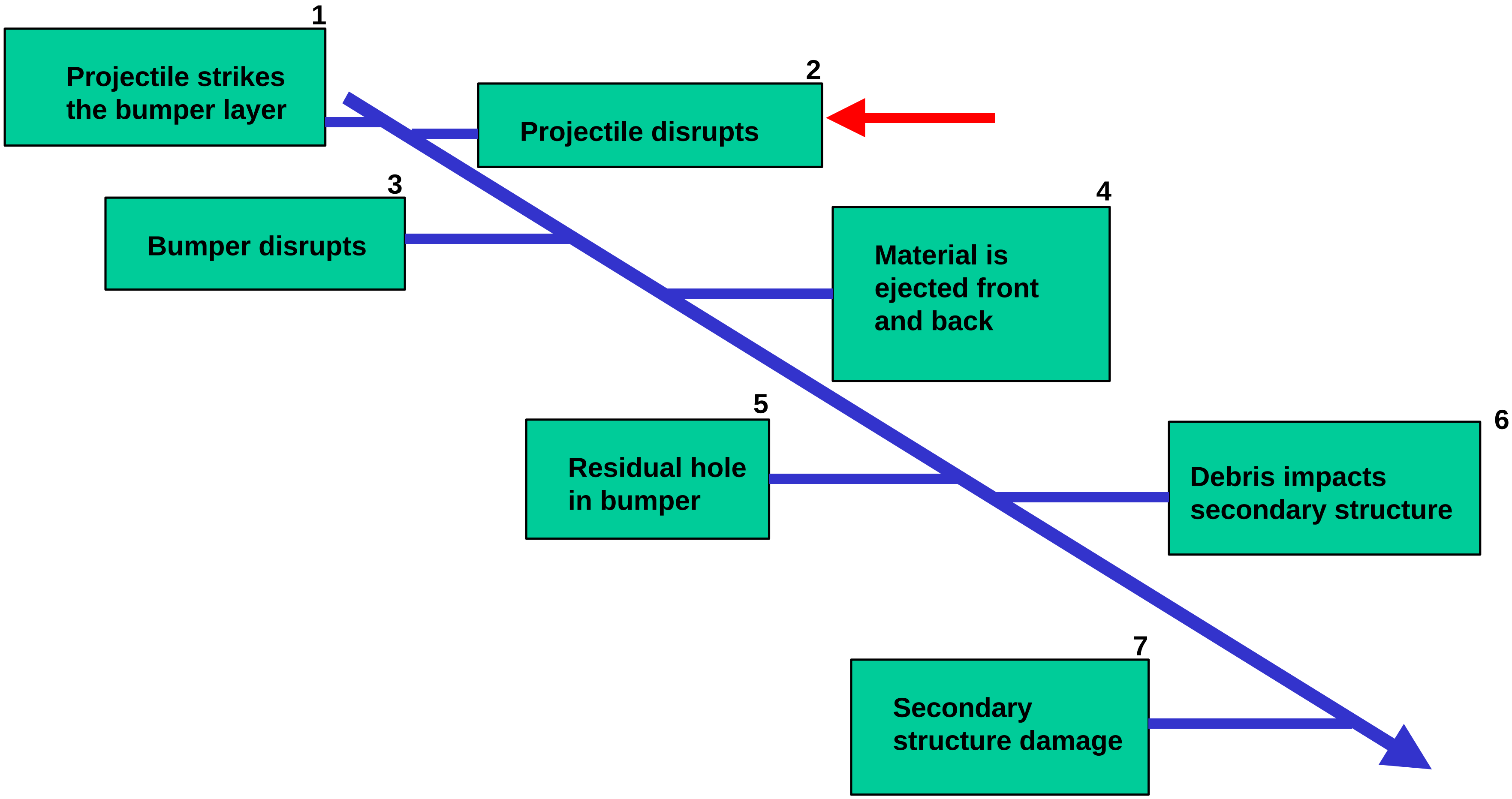
The necessary information is a mix of qualitative and quantitative.

"Validation data".

Consider validating the microscale with laboratory hypervelocity impact experiments.



Levels of phenomenology at the microscale:



Predict secondary damage

Further levels of phenomenology in this experiment:

Projectile disrupts

Temporally, spatially varying compressive waves

EOS, constitutive behavior in compression: Hugoniot states, two and multi-wave structures, compressive strain- and strain-rate dependent strength effects

Wave propagation, attenuation, reflection from free surface

Correct wave speeds, non-planar waves; rise-time effects at reflections; hydrodynamic and dissipative wave attenuation, time-dependent spall, nucleation and growth of void; shear localization

Release states form and interact: spall, fracture, fragmentation, melting, boiling

Time-dependent spall, ejecta, phase transitions; kinetics; thermal localization and trapping; non-classical viscosity; damage nucleation and growth; statistical breakup phenomena

The parameter space for these experiments:

"Experimental" parameters:

- •Impact location (2)
- •Velocity vector (3)
- •D, d (2)
- •Projectile geometry (1)
- •Materials (none)

Total = 8

"Internal" parameters:

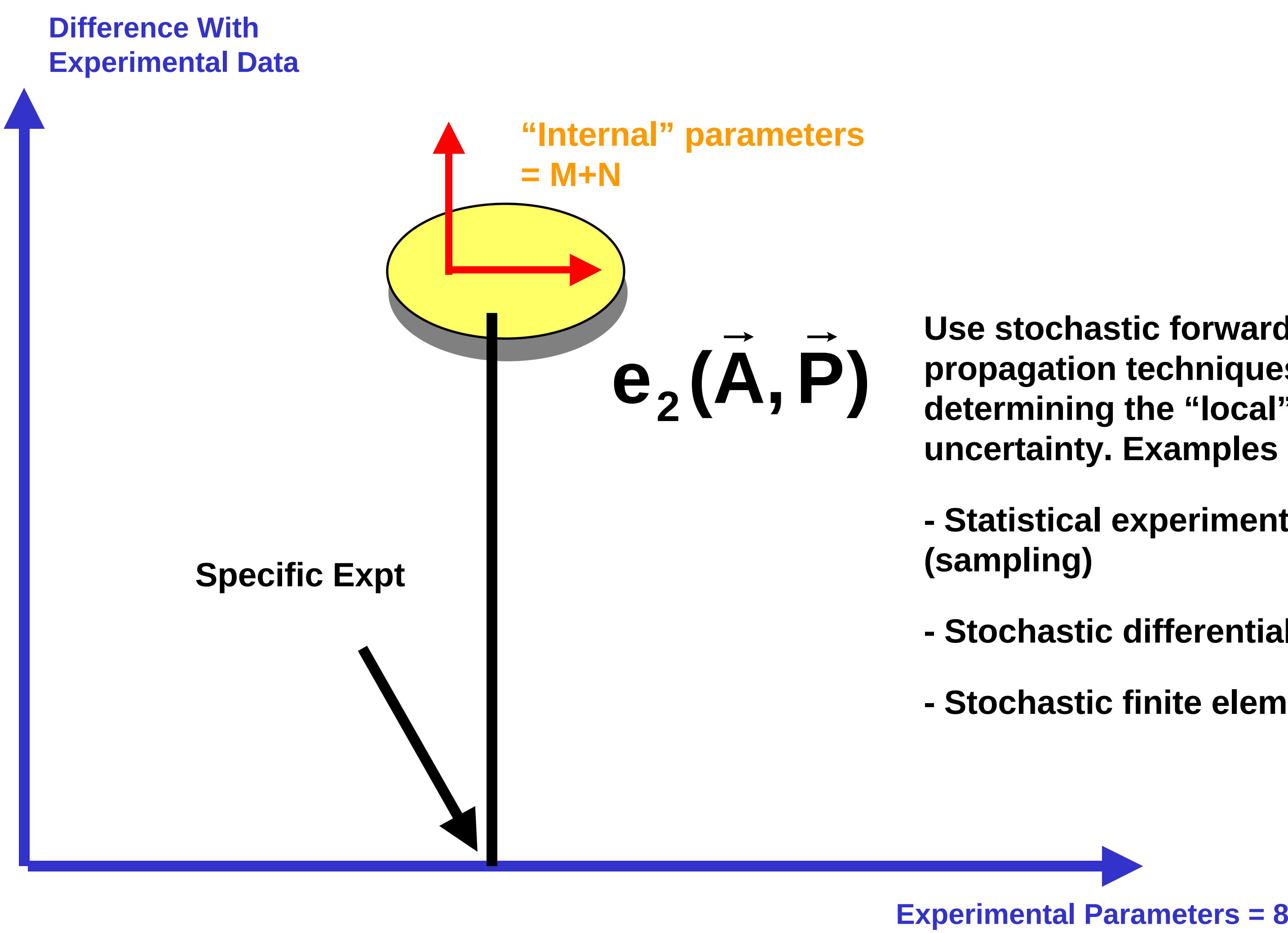
- •Hydrodynamic parameters (H)
- •Material Models (M)

Total = H+M

"Uncertainty" Metrics:

- •Time-resolved data in witness material (PVF gauges)
- •Time- and spatially-resolved debris cloud data (radiography)
- •Target recovery and inspection

Parameter versus uncertainty space



Use stochastic forward propagation techniques for determining the “local” uncertainty. Examples include

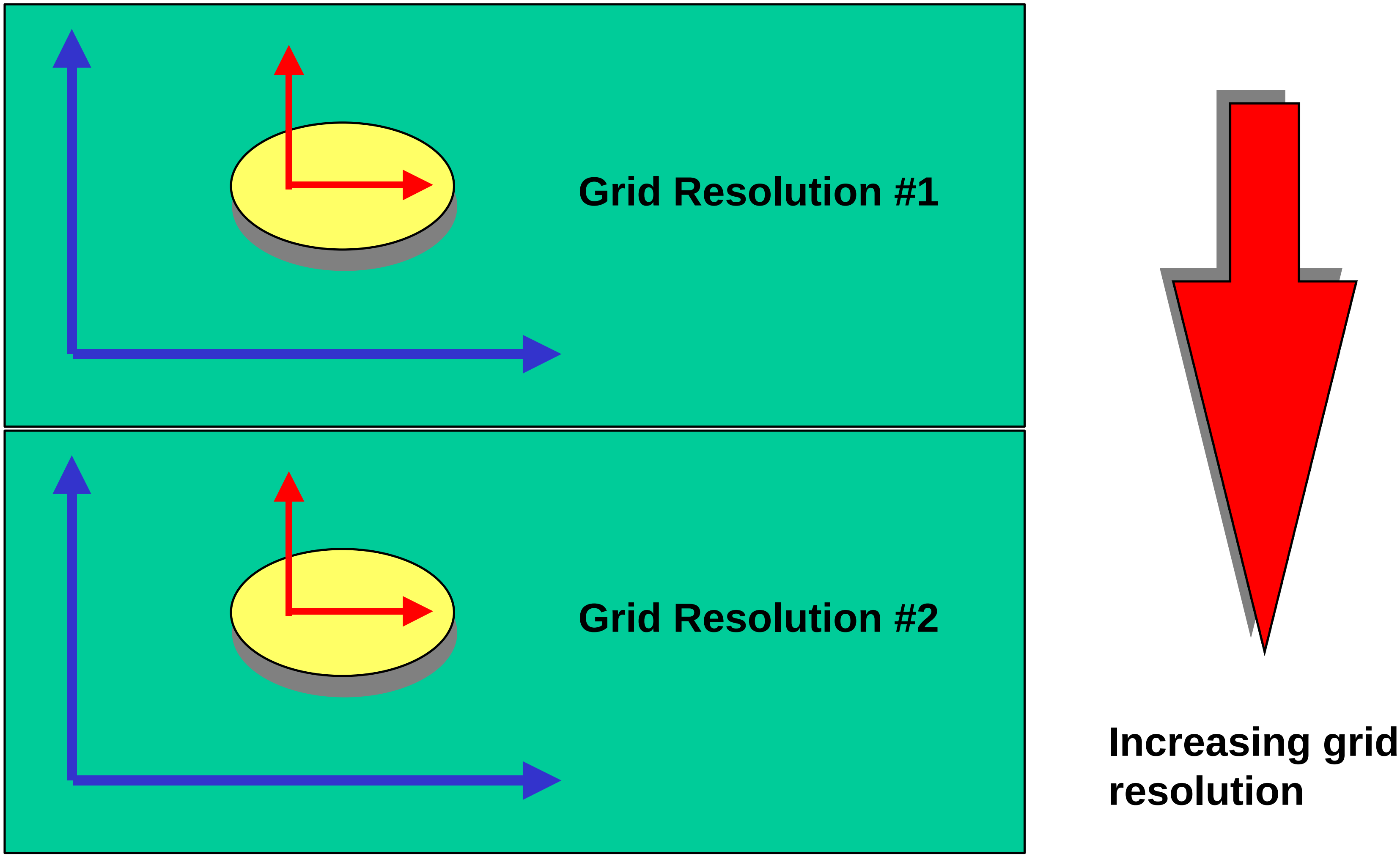
- Statistical experimental design (sampling)

- Stochastic differential equations

- Stochastic finite elements

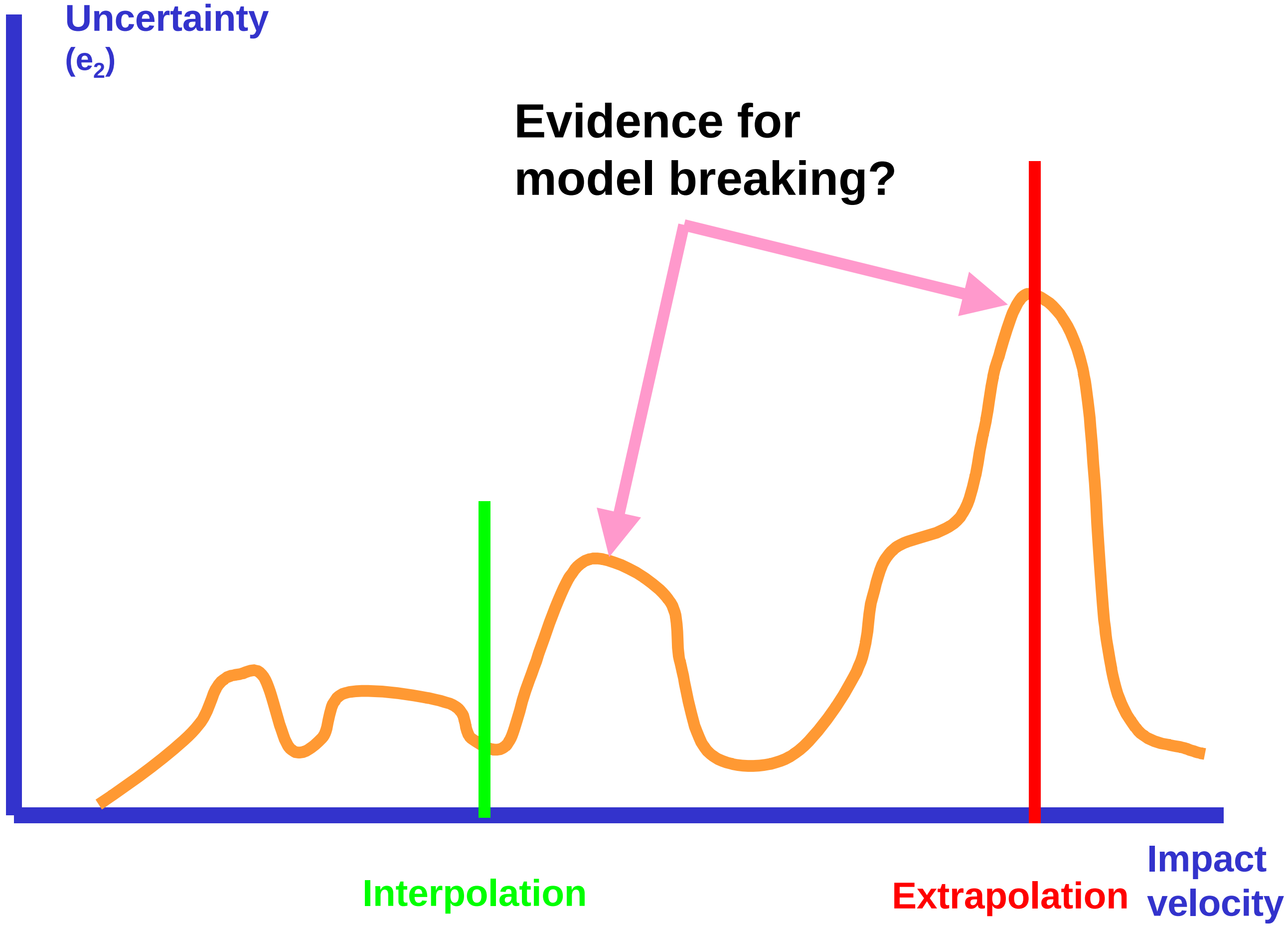
Sandia National Laboratories

One way to worry about grid resolution:

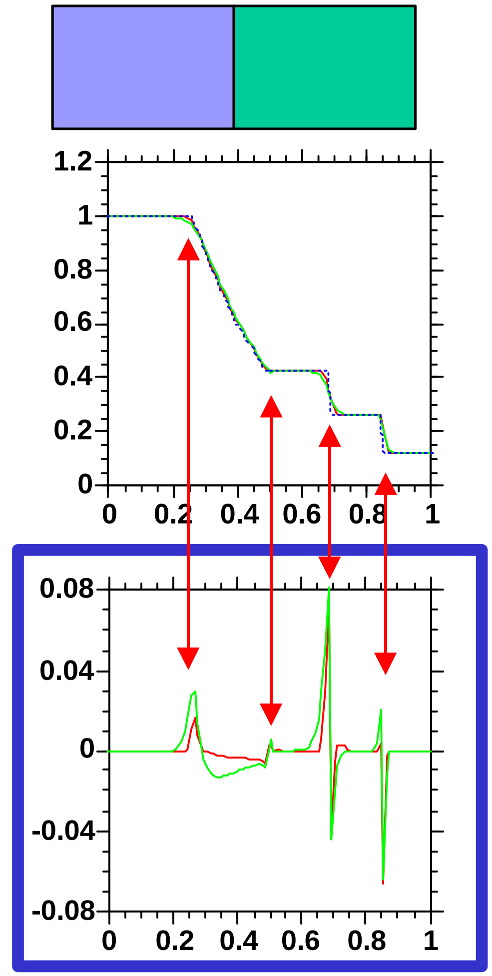


Increasing grid resolution does not mean uniform refinement (ALE, adaptivity, geometry constraints). Algorithm parameters controlling dynamic grid resolution are included in the internal parameters.

A thought experiment: projecting the uncertainty



Sod Problem



Thanks to Kamm and Rider

Some speculation and reasonable opportunities for research: Uncertainty as a spatial random field

I. Let the uncertainty be represented by a (M+H+8) dimensional spatial stochastic process

For example, we end up worrying about properties of something like the <u>variogram</u>:

$$\mathrm{var}[e_2(\vec{A}_j,\vec{P}_j) - e_2(\vec{A}_i,\vec{P}_i)] = 2\gamma((\vec{A}_j,\vec{P}_j) - (\vec{A}_i,\vec{P}_i))$$

Then, we can develop predictors for U at other points, such as the BLUP (Best Linear Unbiased Predictor):

$$\hat{e}_2(\vec{A}_0,\vec{P}_0) = \sum_{i=1}^{n} \lambda_i \hat{e}_2(\vec{A}_i,\vec{P}_i)$$

Or (μ = mean of random field, S = fine structure, and ε = "measurement error":

$$\tilde{e}_2(\vec{A},\vec{P}) = \mu(\vec{A},\vec{P}) + S(\vec{A},\vec{P}) + \varepsilon(\vec{A})$$

Comments:

See J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn (1989), "Design and Analysis of Computer Experiments," Statistical Science, Vol. 4, No. 4, 409-435; N. Cressie (1988), "Spatial Prediction and Ordinary Kriging," Mathematical Geology, Vol. 20, No. 4, 405-421.

Do we need another framework other than probability to do this?

What is the appropriate way to partition this random field among the experimental parameters and the internal parameters? Does this question even make sense?

What structure do we require on the projected random field $e_2(A,P)$ to facilitate piecing together the various uncertainties?

Speculation: Sensitivity coefficients

II. Sensitivity studies define which of the H+M+8 parameters is most important. Probabilistic evaluation of the sensitivities is of interest.

Is parsimony really true?

Does the sensitivity structure projected onto the internal parameter space remain invariant as the experimental parameters alone vary? If no, does this imply model invalidity?

Does the sensitivity structure remain invariant over grid variations?

Don't assume that the parameters are all uncorrelated. Then we need "interaction" coefficients.

The literature on sensitivity analysis is huge. See M. D. McKay (1995), "Evaluating Prediction Uncertainty," Los Alamos Report, LA-12915-MS for one approach that we are using in "DDACE."

Speculation: Model calibration

III. Some understanding of e_2 should lead to improvement in the model. "Calibration" reduces e_2 locally in the application space by optimizing the internal parameters.

How does the calibration vary with the experimental parameters?

Comments:

A Bayesian approach can be applied to the formal study of improving model uncertainty in the presence of parameters derived via comparison with experimental data. See, for example, K. M. Hanson (1998), “A Framework for Assessing Uncertainties in Simulation Predictions,” Los Alamos preprint.

Statistically rigorous comparisons between uncertain calculations and uncertain data with the intent of providing code validation are the subject of a recent tutorial report by R. G. Hills (1998), “Statistical Validation of Engineering and Scientific Models: Background,” Sandia National Laboratories Contract Report.

This question also leads to the use of “surrogates” for studying parameter calibration, as well as other optimization questions associated with code uncertainty. Consider the important work A. J. Booker, et al (1998), “A Rigorous Framework for Optimization of Expensive Functions by Surrogates,” Boeing Shared Services Group Report, SSGTECH-98-005.

Other papers that the reader might find of interest are D. D. Cox, J. Park, and C. E. Singer (1996), “A Statistical Method for Tuning a Computer Code to a Data Base,” Rice University Department of Statistics Report 96-3 and M. B. Beck (1987), “Water Quality Modeling: A Review of the Analysis of Uncertainty,” Water Resources Journal, Vol. 23, No. 8, 1393-1442.

Speculation: Structural (Model) uncertainty

IV. Is there anyway to deal with “structural” uncertainty?

A Bayesian structure can be developed for considering structural uncertainty. See D. Draper (1995), “Assessment and Propagation of Model Uncertainty,” J. R. Statist. Soc. B, Vol. 57, No. 1, 45-97.

This involves developing posteriors via conditioning over the space of models, a rather hopeless endeavor on the face of it. Additional structure might make this more feasible.

Model uncertainty is often treated in multi-physics code through the introduction of tuning parameters. If a code (sub)model is built out of sub-submodels:

$$\mathbf{M} = \sum_j \mathbf{M}_j$$

Uncertainty about the overall model is then treated by modifying this equation to:

$$\mathbf{M} = \sum_j \alpha_j \mathbf{M}_j$$

Add (α_1, …, α_m) to the parameter list and proceed as before.

Sandia National Laboratories

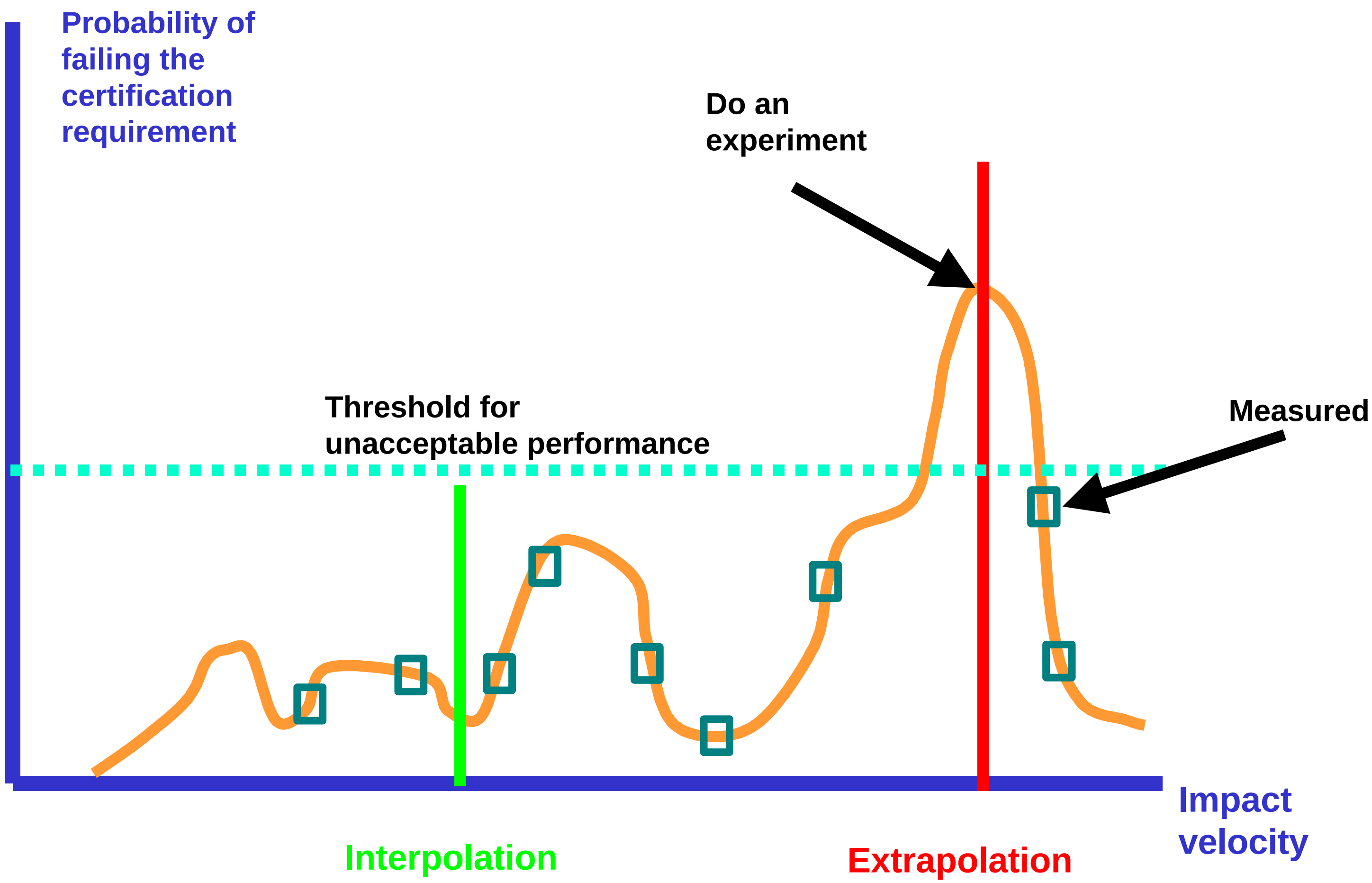
Speculation: Is probability appropriate for this discussion?

How do we treat “variability” in the assumed distributions of the parameters to do experimental design, sensitivity analysis, and forward propagation?

Is probability the canonical way to capture “uncertainty?”

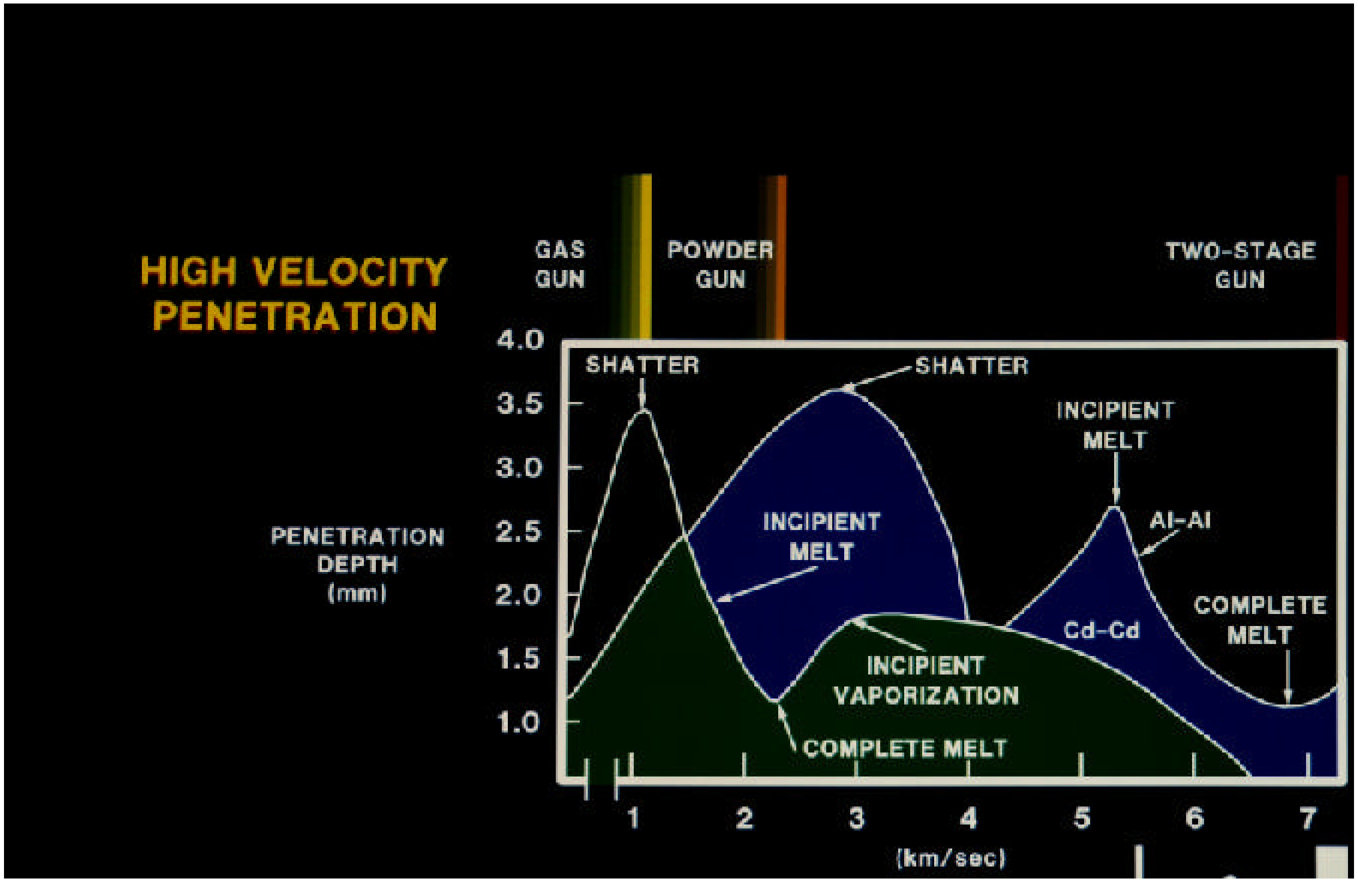
> **Is saying “I don’t know what the value of a parameter is” the same as placing a probability distribution on it?**

Speculation: "Certification" or "Accreditation" leads to quantitative "reliability" analysis.



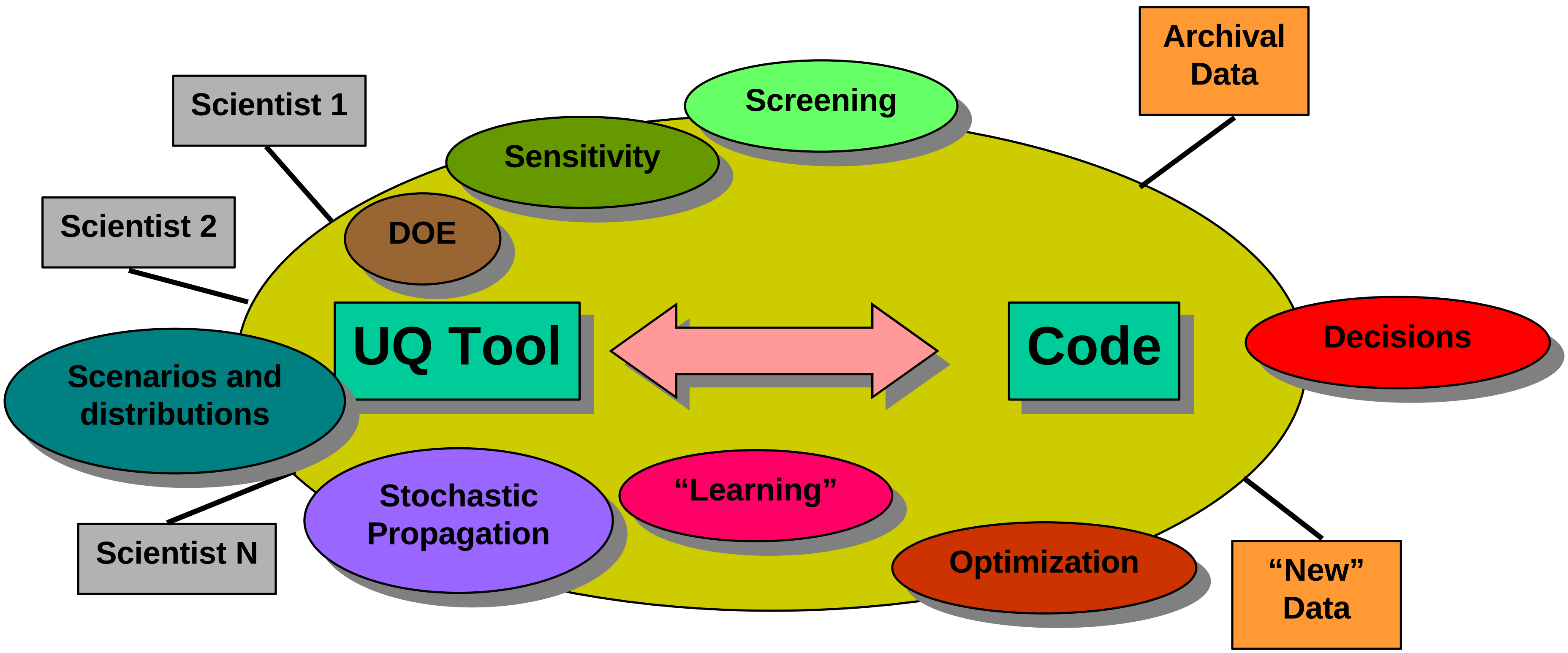
Sandia National Laboratories

There is some connection to reality here.



Sandia National Laboratories

A UQ tool should become part of a conceptual design or decision making environment utilizing computation.

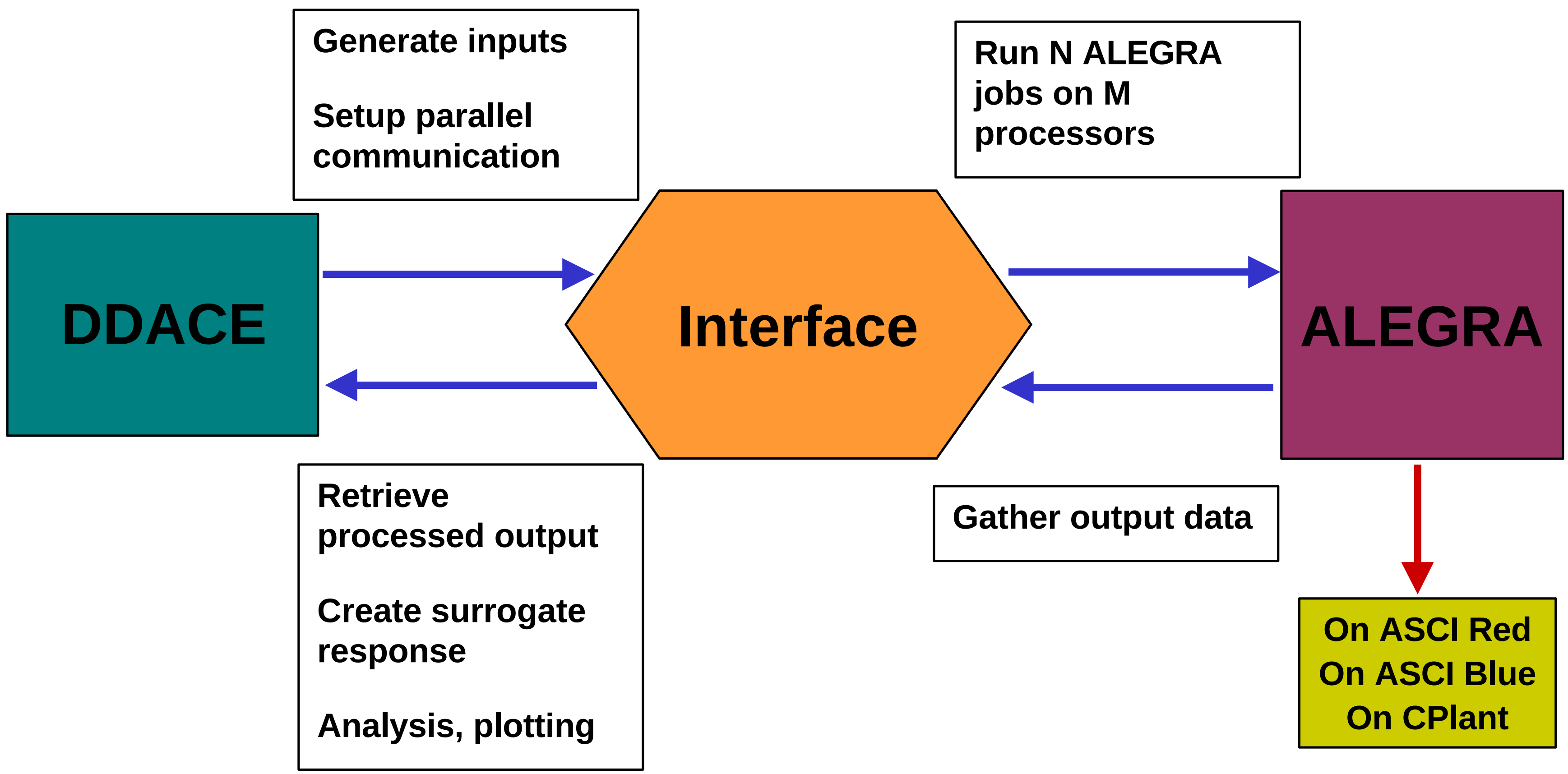


We are undertaking an exploratory development project to prototype a UQ tool for use with our shock wave physics codes.

Sandia National Laboratories

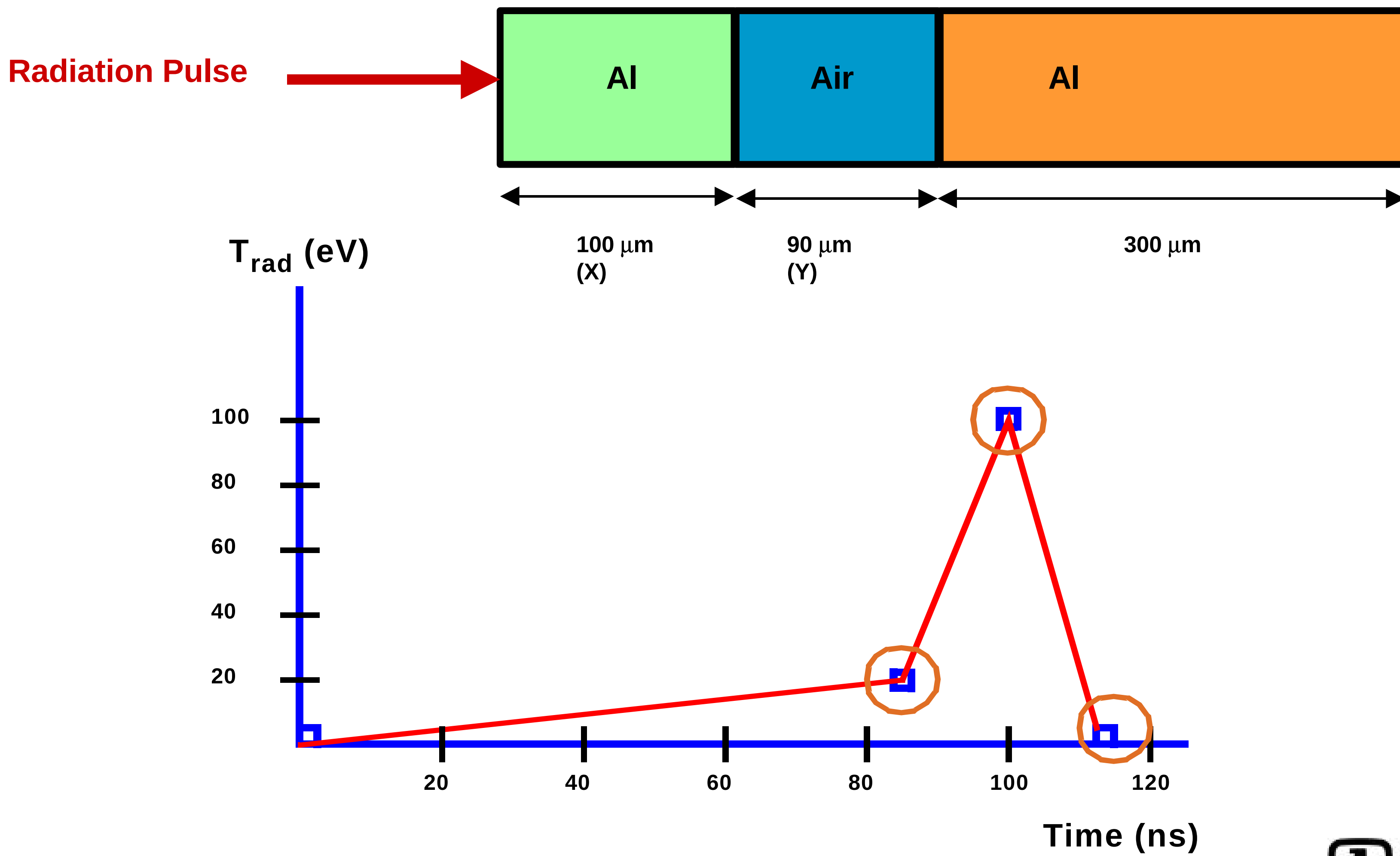
DOOMSDACE: Distributed Object-Oriented Software With Multiple Samplings for the Design and Analysis of Computer Experiments

(Tong, Long, Lehoucq, Plantegenet, Meza)



Sandia National Laboratories

Example: A 1-D rad-hydro calculation



Variability parameters:

Parameters	**#**
Numerical parameters: timestep, artificial viscosity (5% variations)	3
Geometry parameters: x (50-200 μm) and y (50-200 μm)	2
Material parameters: gas density (variation to be determined)	1
Radiation pulse parameters: (t_i, $T_{rad,i}$), i=1,2,3 (10% variations)	6
Radiation algorithm parameters: ?? (to be determined)	2
XSN opacity parameters: ?? (to be determined)	2-4
Total:	~18

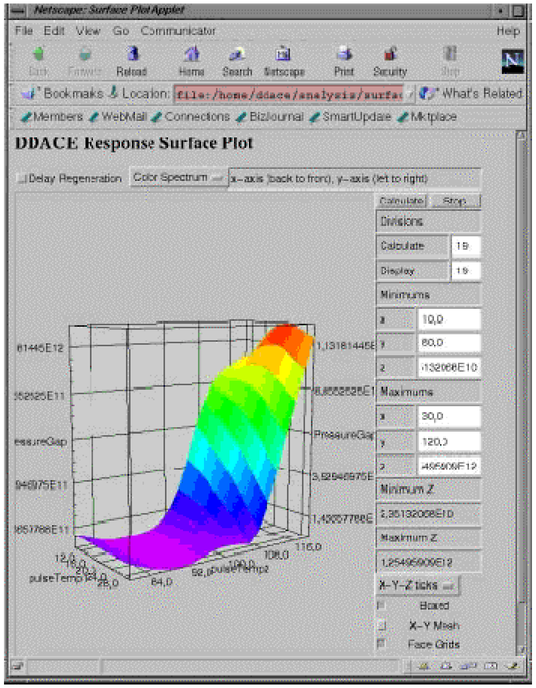
A recent design study shows what is on our mind.

Work performed by Kevin Long.

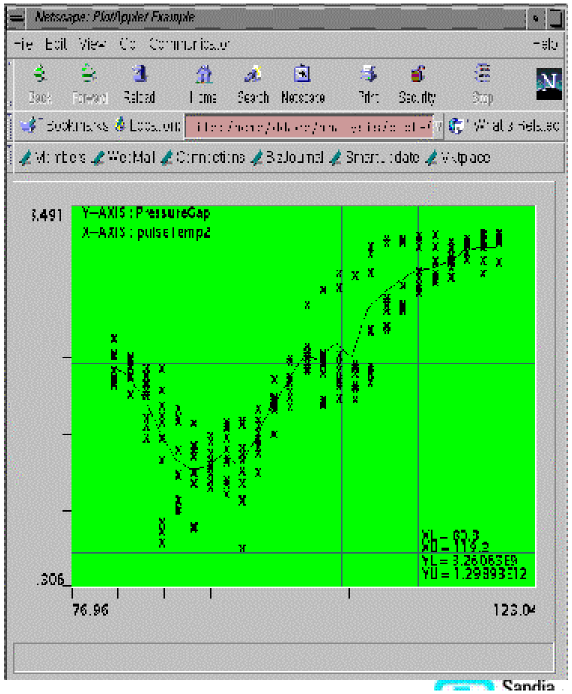
Experimental Design - 250 design points (LHS?).

Run on 32 processors of Elmo (a California SGI). Each function evaluation takes about 1/2 hour or more on the Ultra 2 in my office. You do the arithmetic related to serial running of this. Think about the scaling to 2000 processors on ASCI Red.

Six hours wall clock time for the total experiment.



Response surface construction for "quality metric" vs two pulse parameters.



"Quality metric" vs most sensitive pulse parameter.

Sandia National Laboratories

In conclusion:

"Summary: Computers Are Here To Stay. They Endanger Thought, Language, Science, and the Survival of Man. Like Any Other Dangerous Tool, They Should Be Put Under Strict Controls."

Clifford Truesdell, "The Computer: Ruin of Science, Threat to Man" in An Idiot's Fugitive Essays on Science