UCRL-PRES-229957

LLNL ASC V&V Strategy

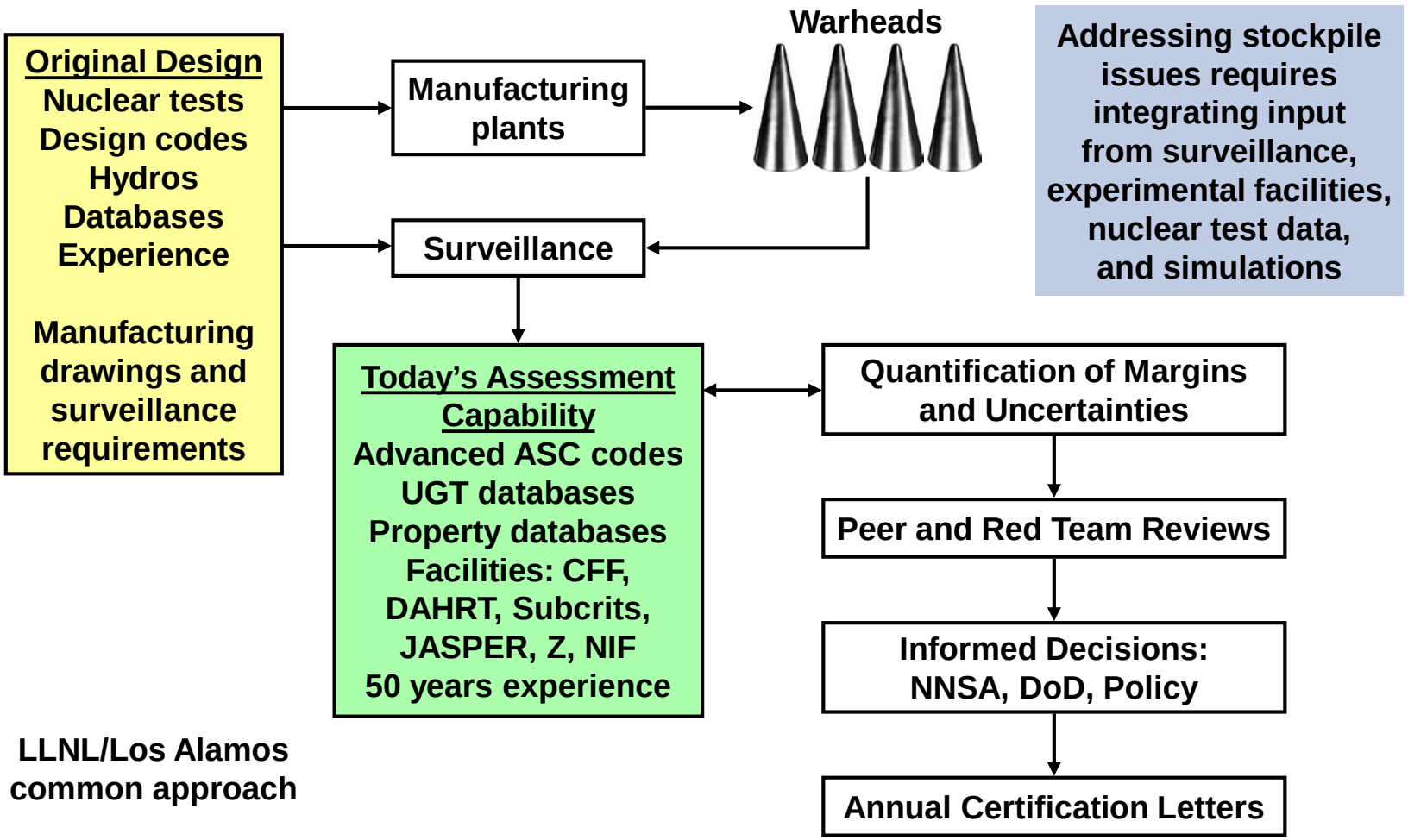
Dr. Joseph A. Sefcik

Associate Program Leader
Defense and Nuclear Technologies
Lawrence Livermore National Laboratory

Salishan Conference
April 22, 2007

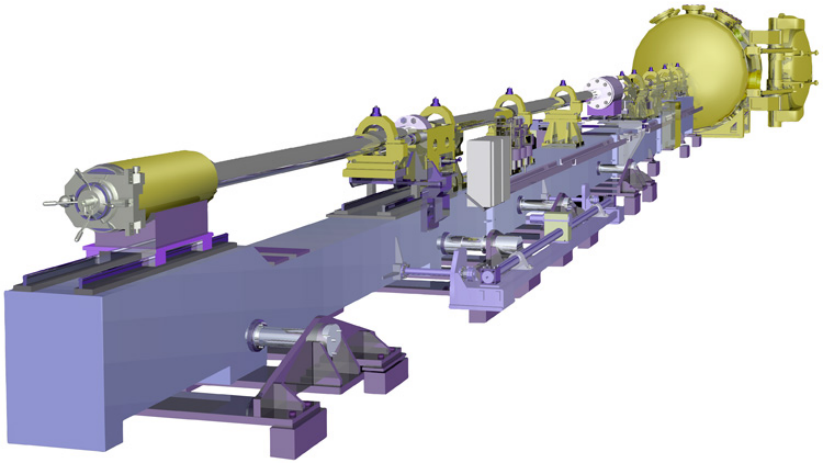
This work was performed under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

QMU is best understood from the perspective of the overall Stockpile Stewardship enterprise



Assessments of warhead performance “at target” involves both reliability and QMU

- **Standard industrial reliability techniques are applied to the myriad components of the weapon system to determine the probability that the system will arrive at target and fire the device**



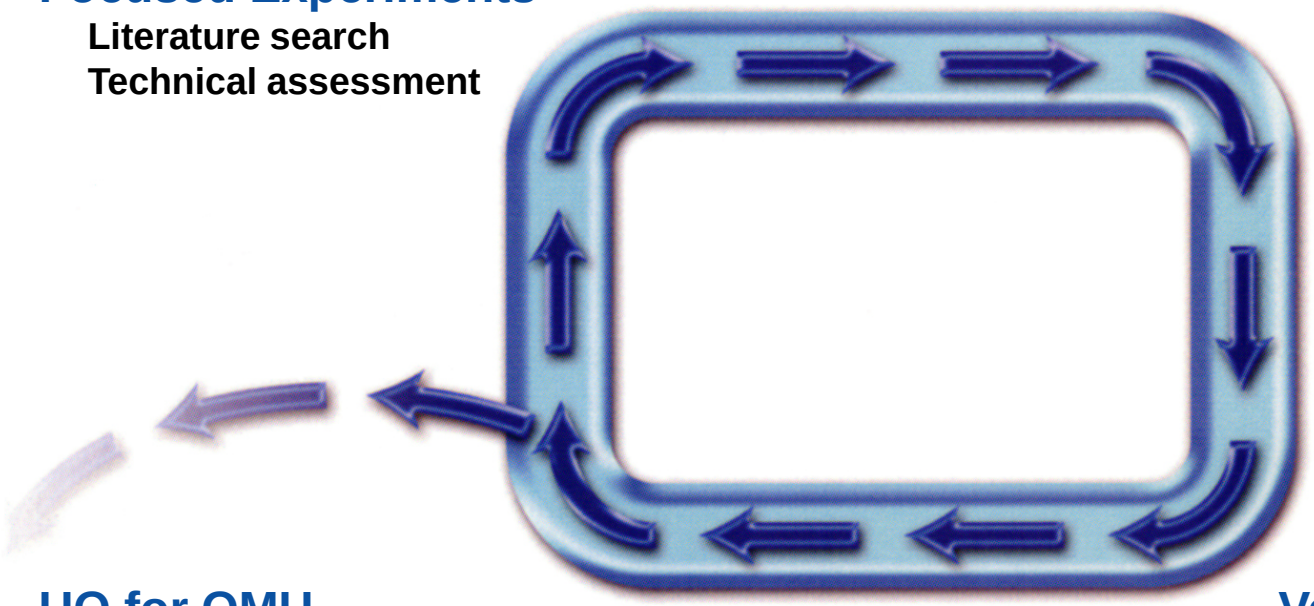
- **The performance of the device is assessed based on fundamental physics data taken from experimental measurements**

The V&V Program supports QMU and the ASC roadmap to predictive capability

- Apply Software Quality Engineering as an integral component of increasing the confidence in ASC codes
- Identify and assess adjustable code parameters with PDFs provided by ASC and Campaigns
- Apply sensitivity analysis to guide resource investments
- Develop methodologies to quantify uncertainties (competing methods will be fostered) and use them to assess weapon performance
- Assure these methodologies meet the criteria for numerical convergence

V&V will provide the discipline and the structure for the UQ in QMU

V&V is an essential part of the ASC product cycle, and an important interface to Science Campaigns



New Capability (new models)

Focused Experiments

- Literature search
- Technical assessment

Software Quality Assurance SQA (SQE)

- Version release
- Regression testing
- Change control
- Code coverage
- Code A ´ = Code A

UQ for QMU

- Assessment of code
- Physics and numerics
- Code A ´ = Code A
- Method A = Method B

Validation Tests

- Comparison with experiments
- Integral test problems
- Regression testing
- Code coverage

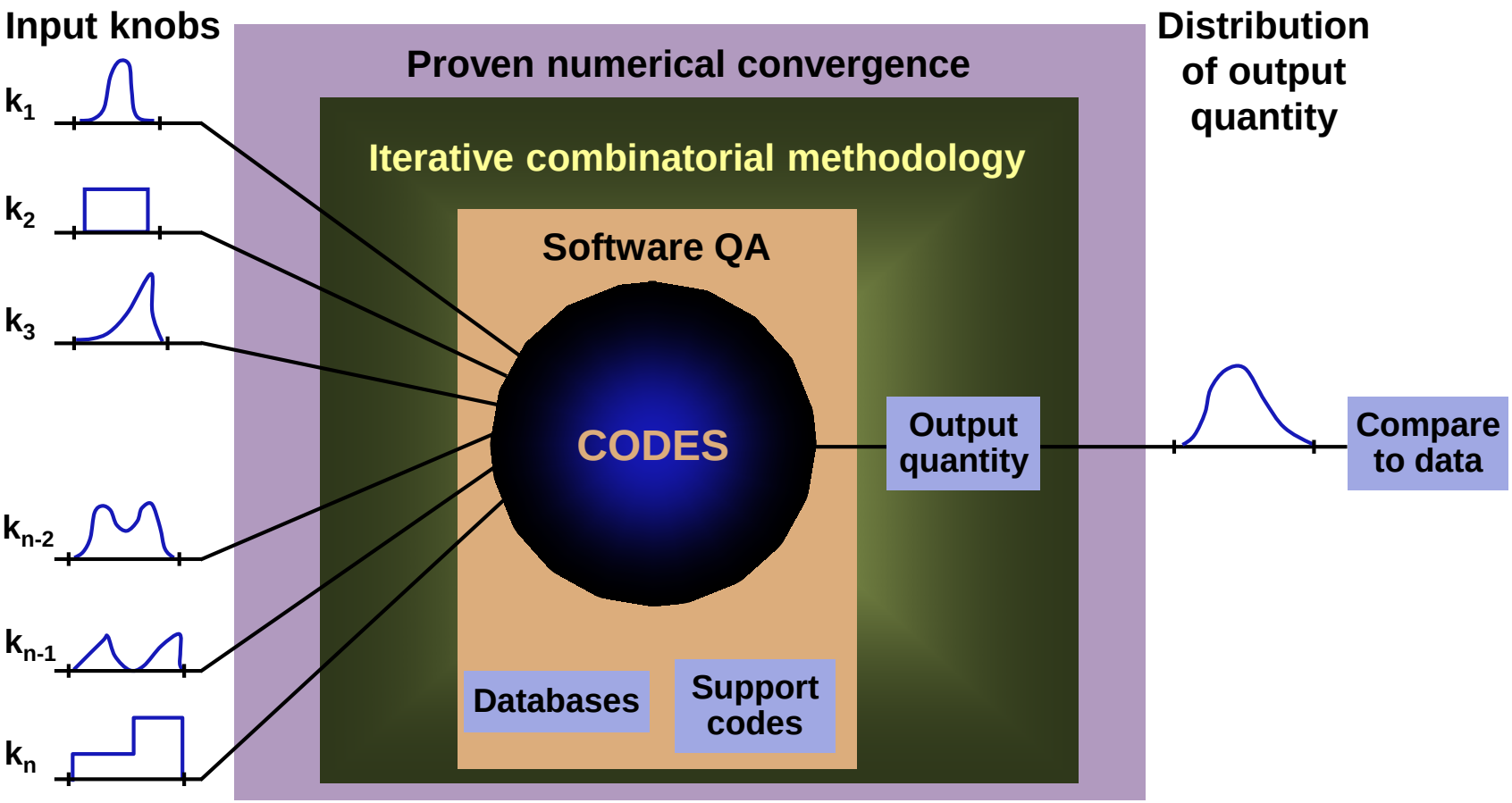
Verification Tests

- Analytical test problems
- Numerical convergence
- Code coverage
- Regression testing
- Code A ´ = Code A

Necessary (and probably sufficient) conditions for achieving predictive capability

- Codes with assured software quality
- Problems run with demonstrated numerical convergence
- Complete identification of all parameters that can be "adjusted" (free parameters)
- Development of distribution functions for the "free parameters"
- An iterative combinatorial methodology that samples the hyperspace of free parameters and uses the code as an "operator" to project a quantity of interest
- Capacity computing sufficiently large to accommodate all of the above in 3D
- Program to elucidate underlying physical phenomena via experiment, theory, and simulation
- An ability to explain the anomalies in the experimental database

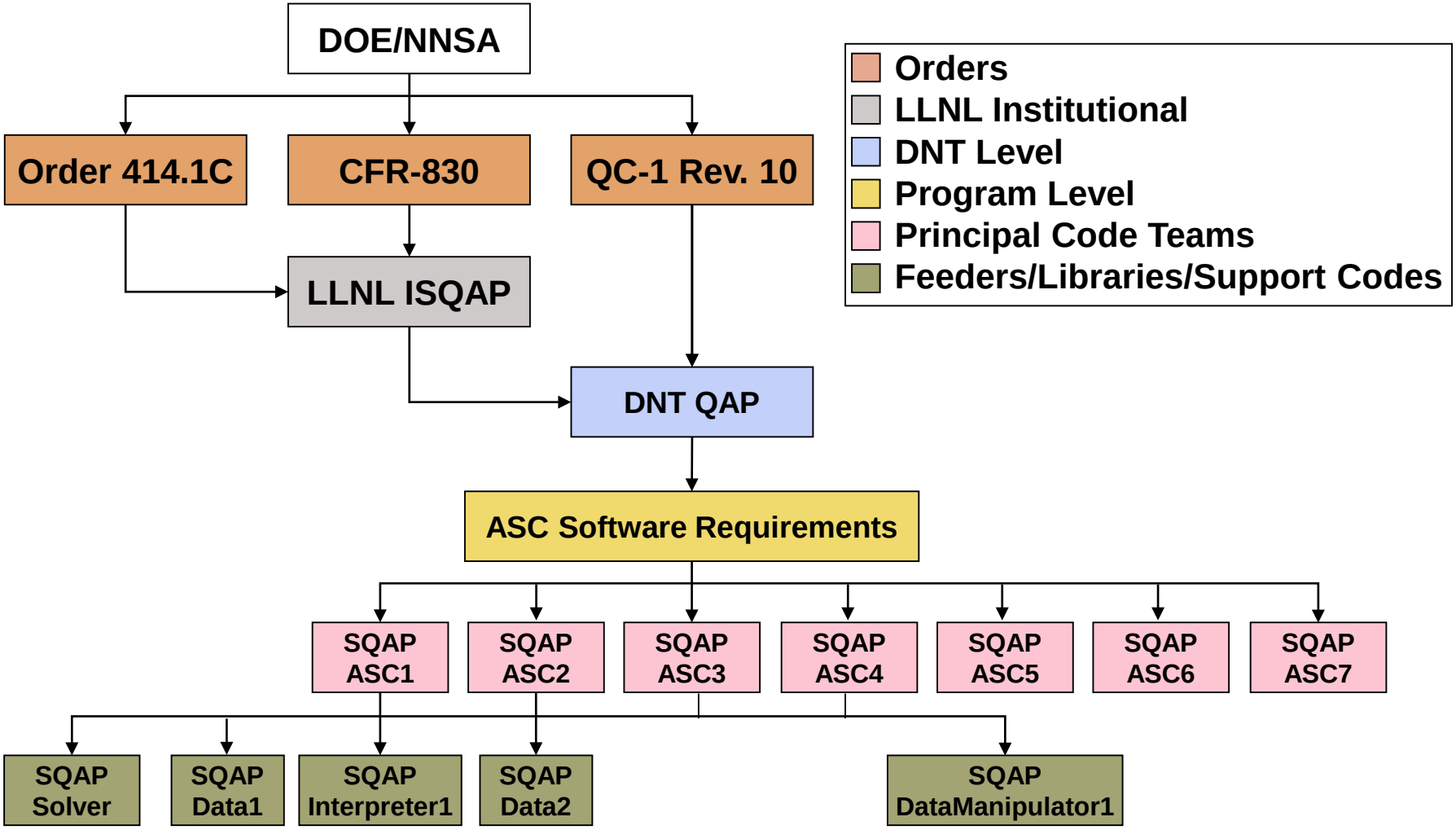
We will quantify probability distributions in key adjustable code parameters and their impact on weapon performance



The codes have hundreds of adjustable parameters; sensitivity analysis will down-select the important ones; UQ techniques can create the final output distributions

LL-DNT-U-2007-012184.7

We are moving the full suite of ASC codes and tools to compliance with DOE Orders and Policies



We are working to demonstrate QC-1 (DOE Weapons Quality Policy) compliance in FY08

- Issue tracking (SQAP)
- Automated regression testing (STP)
  - Function/Statement coverage using commercial tools
- Configuration management (SCMP)
- Disaster recovery (SDRP)
- Coding standards
- Library standards
- ASC V&V SQE embedded staff
- Continuous process improvement
  - Commercial error checking software

We have over 30 code packages and several million lines of code to address

We know where the current “free parameters” are—they are in the current codes

- If a designer can input a parameter change into the deck that changes an output, then that parameter is a free parameter
- Numerical parameters (e.g., mesh size, rezoning settings, mesh “stiffeners”, etc.) can play in the game, but can be “studied” to convergence
- All free parameters have a distribution function (point values or ranges with shapes, means, moments, etc.)
- The multiple free parameters and their distributions form a hyperspace of inputs and the code (as operator) projects that hyperspace onto an axis of interest

Fortunately, the physics of our devices does not exhibit chaotic behavior (barring obvious errors in Monte Carlo statistics)

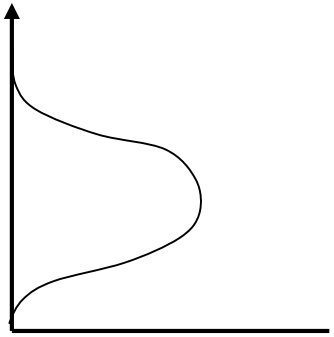
LL-DNT-U-2007-012184.10

Complex computational simulations generally consist of aleatory and epistemic input variables

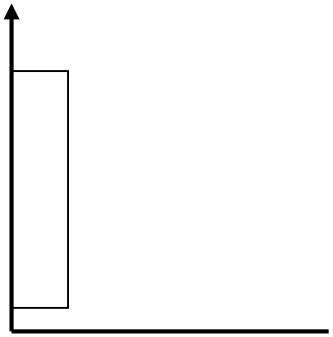
- **Aleatory (Latin *"alea"*—throwing of dice)**
- **Epistemic (Greek *"Episteme"*—knowledge)**
- **An aelatory input distribution must be sampled based on a probabilistic outcome and is random (the dice must be rolled at each input step)**
- **An epistemic distribution is based on scientific assumptions regarding the range of the variable and its distribution**
- **Input parameters for the physics package are purely epistemic (currently)**

This makes the interval of the input variable the most important thing to know when beginning the iterative combinatorial methodology

Aleatory and epistemic distributions generate different kinds of “error bars”

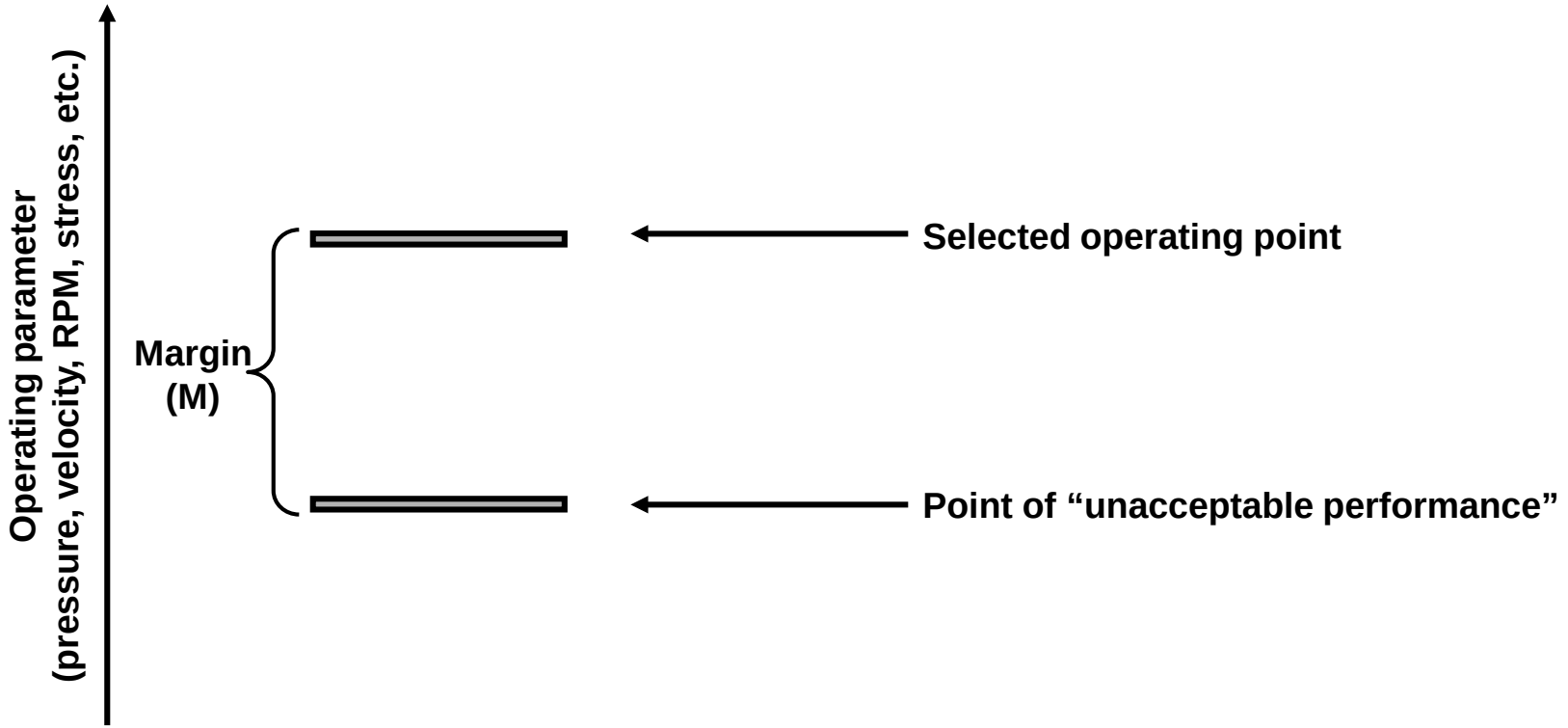


Probabilistic distribution function with some variance based on sampling data

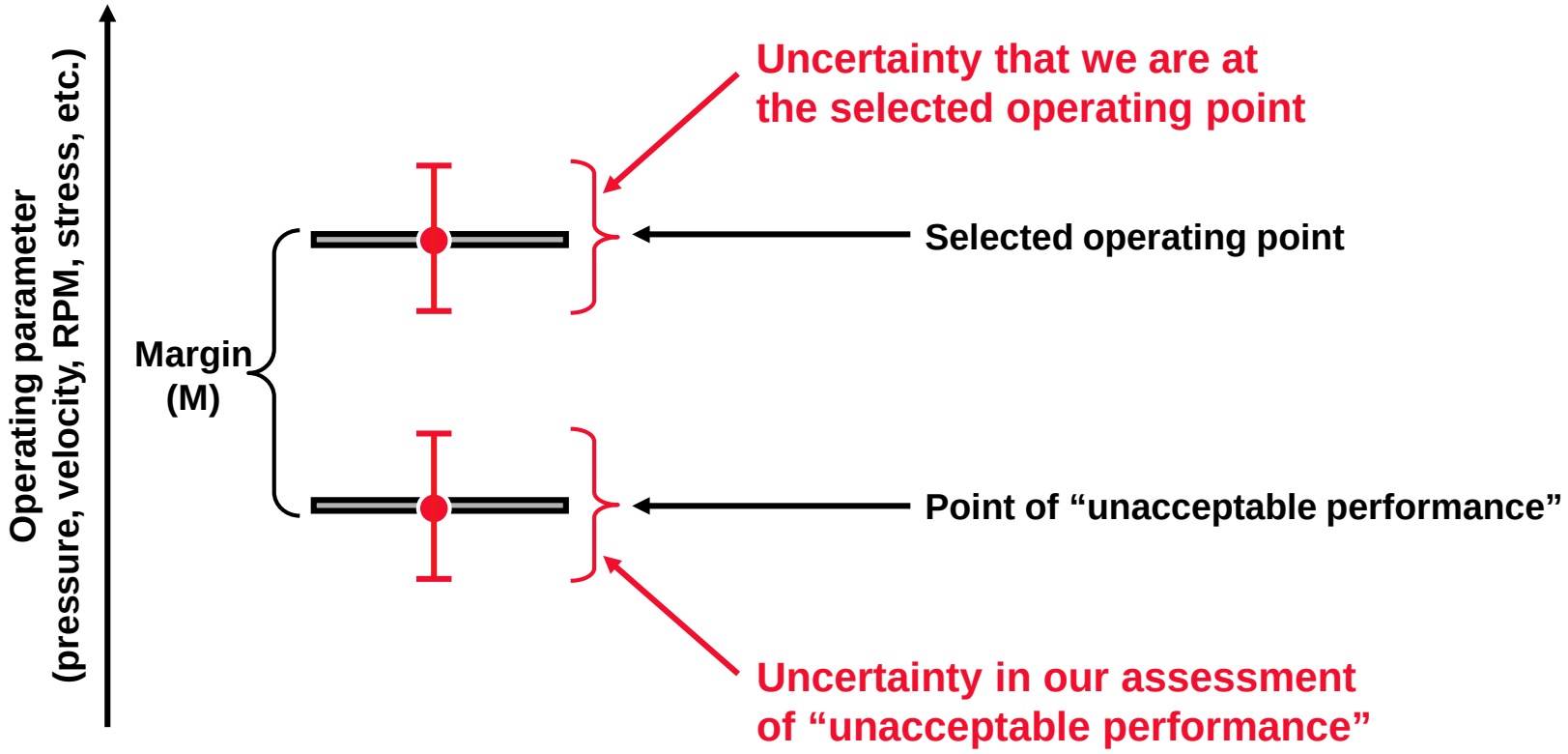


Uncertainty in physics of an adjustable parameter, but a known range beyond which values would be inconsistent with physical reality

Margin is a well-understood concept in engineering

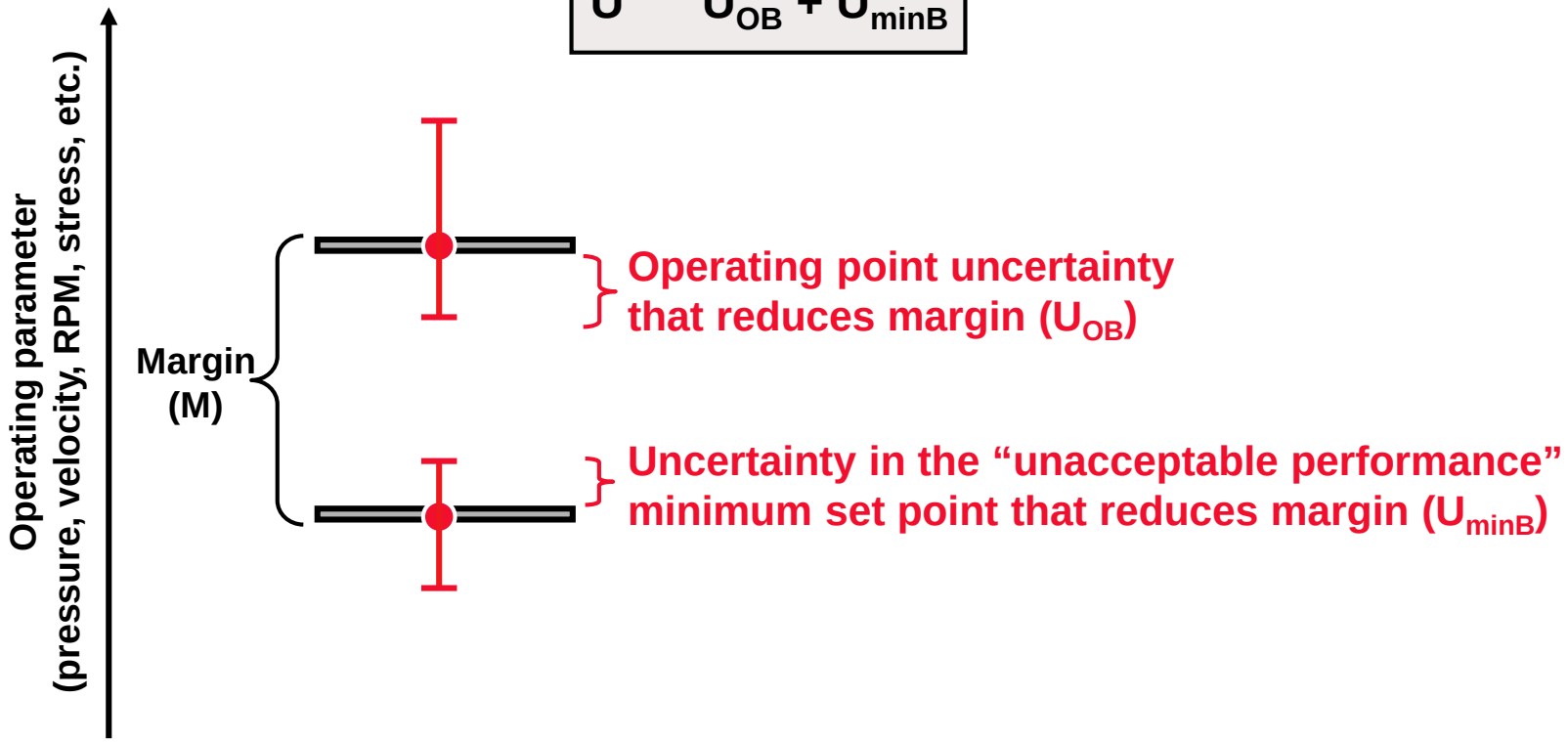


Uncertainty is folded into the picture to reflect confidence, or level of risk, in our decision-making process

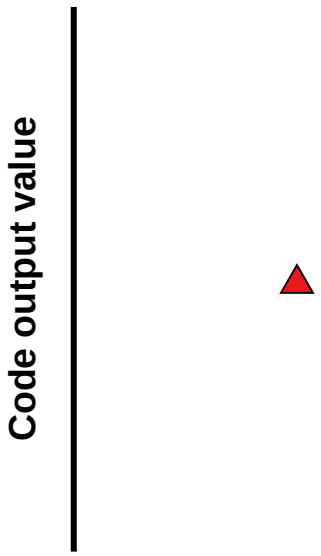


We use M/U to give us an estimate of our risk

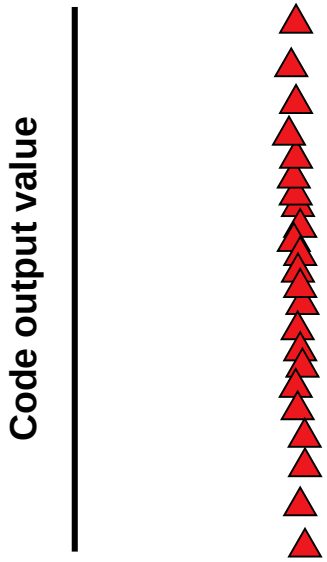
$$\frac{M}{U} = \frac{\text{Margin}}{U_{OB} + U_{minB}}$$



Error bars should not be applied to single calculations, but uncertainties can be assessed for an ensemble of calculations for a model

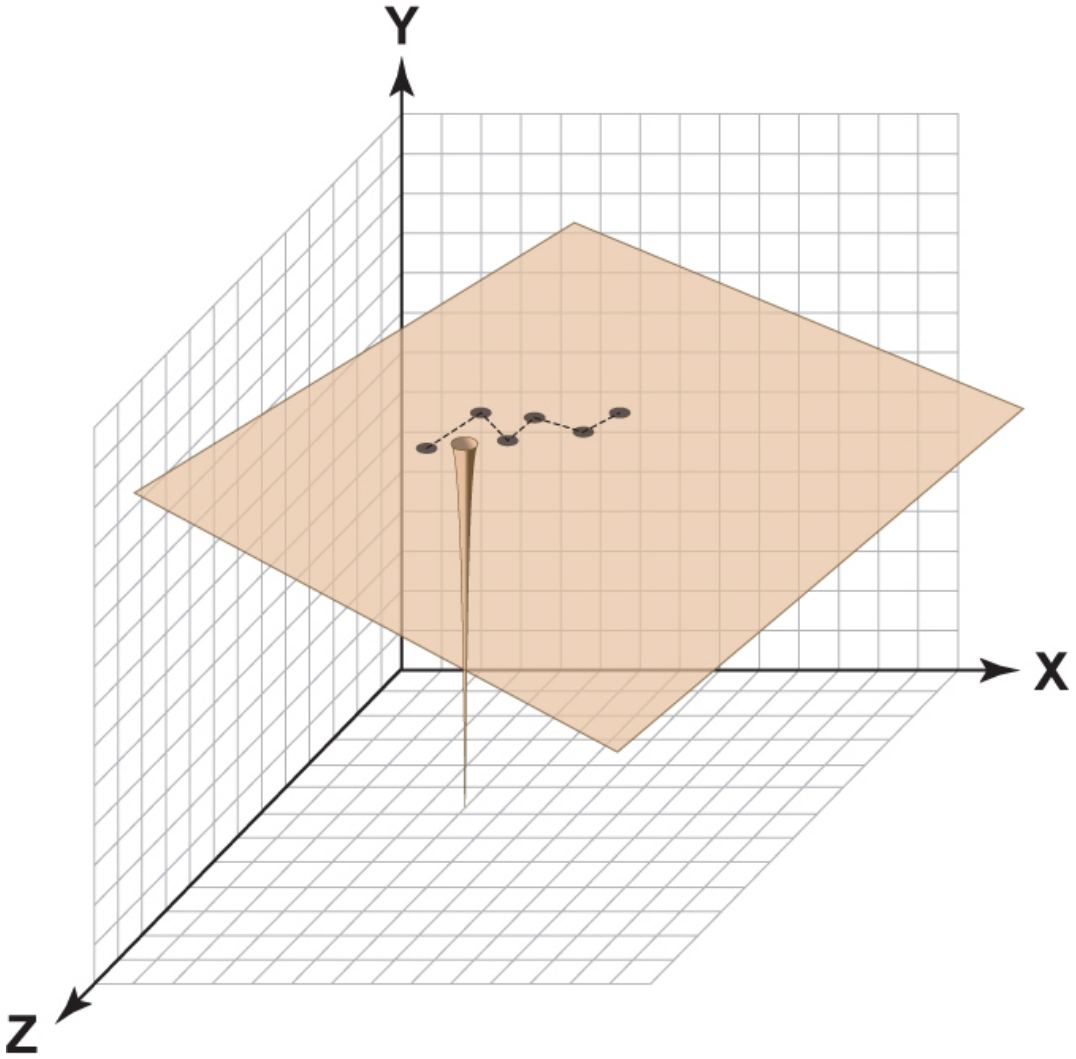


The “error” associated with one computer simulation is essentially the round-off error of the computer hardware

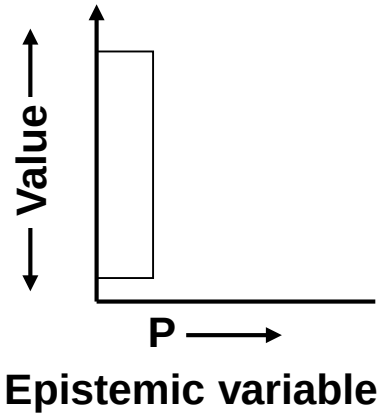


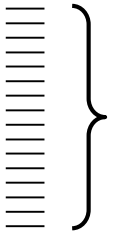
An ensemble of calculations where the input parameters span the available space define the uncertainty associated with a particular model

The potential interactions among free parameters produce non-linearities that could prove disastrous to performance



We would like to robustly locate the wormholes in our hyperspace of variables





Number of sample points = M

Number of free parameters = N

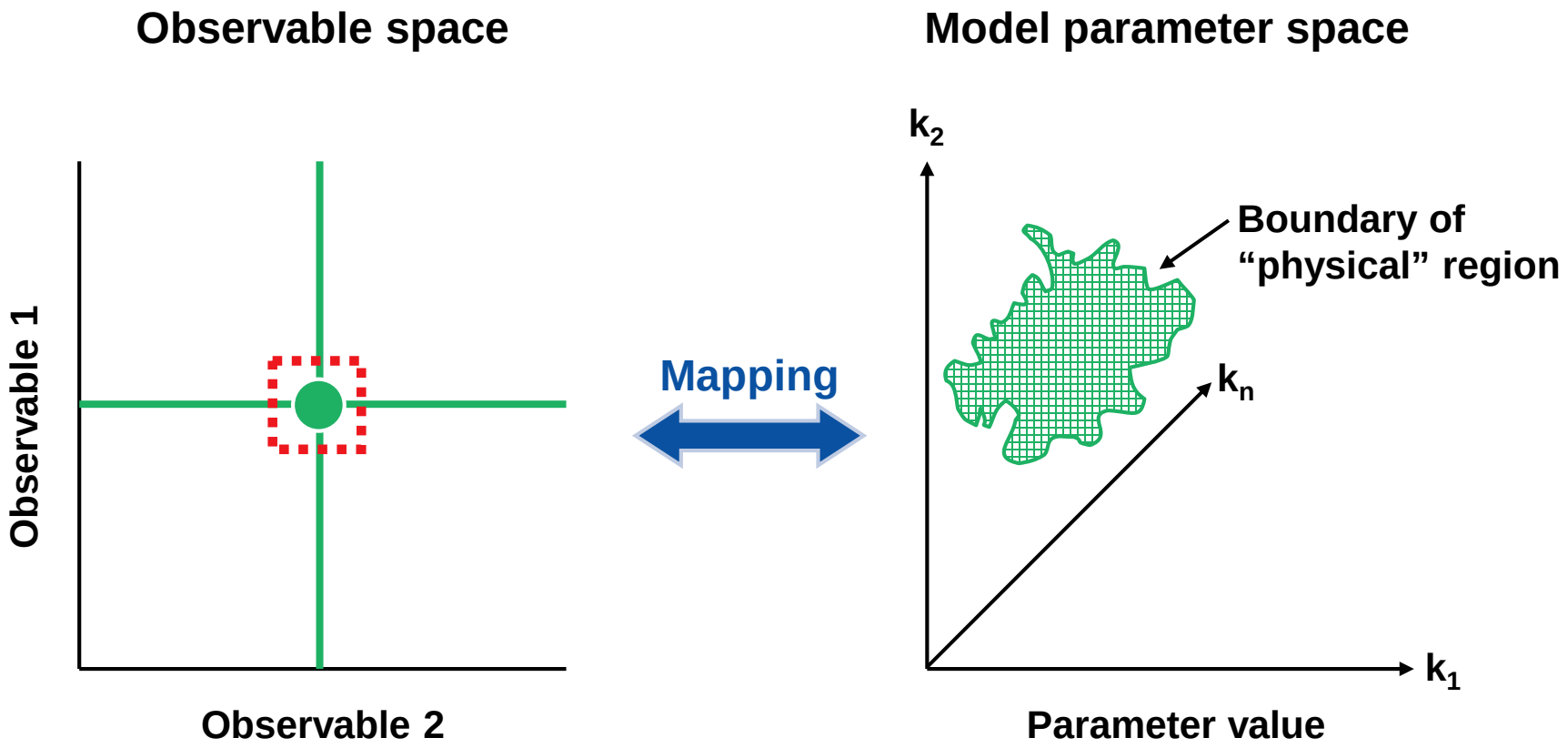
The Curse of Dimensionality

Total number of samples required = $(M)^N$

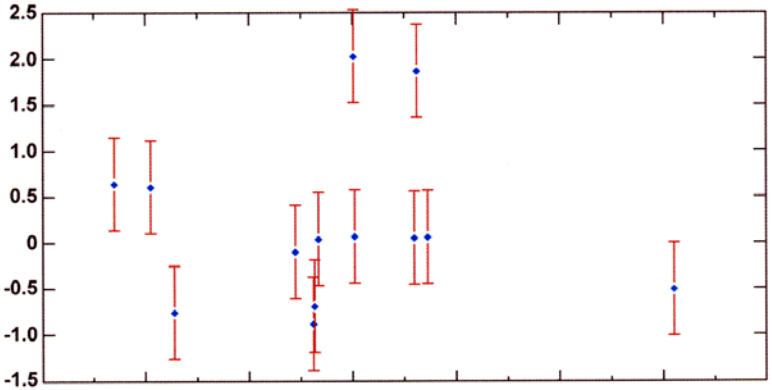
3 samples	2 parameters	$(3)^2$
10 samples	10 parameters	$(10)^{10}$
100 samples	600 parameters	$(10)^{1200}$

Each sample represents a computation in 1D, 2D, or 3D with numerical convergence

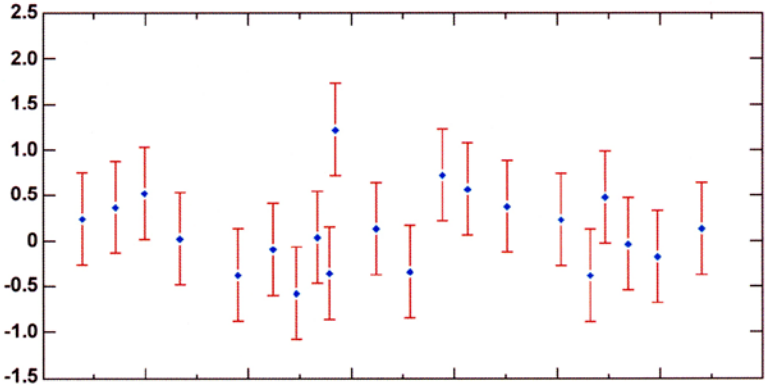
Since calibrated models are used, a single experimental point maps to a volume in the model parameter space



Our 10-year objective is to expand our models and reduce the standard deviation of the set

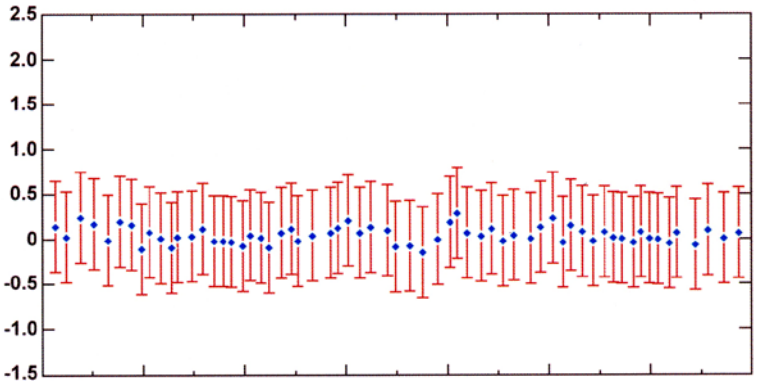


Goal: circa 2010



This will require development of models that move to 3D on Sequoia

Goal: circa 2015



What does predictive capability (sufficiently numerically converged) and in 3D mean for capacity machines?

- 3D design calculations probably require fine zones
- A full device 3D numerical convergence study cannot be run today on purple (100 teraflops)
- A full model set (50–100 data points) in 3D (fully converged) with fine zoning to do parameter surveys will require 10 petaflops of capacity to leave room for others
- Full vetting of the adjustable parameters to match data using stochastic bootstrapping (or MOAT, MC, LHS, etc.) and sampling all of the distribution functions could consume 10–100 petaflops of capacity

We can use sensitivity analysis to cut down on the number of sample free parameters

There are unique “capabilities” that a “capacity” machine can provide to the program

- **There is nothing wrong with performing sensitivity analysis in 1D on one processor**
- **A good 1-D run can be accomplished in 30 minutes**
- **A 120,000 processor machine can run one billion calculations in 6 months**
- **This does not solve the “Curse of Dimensionality” (M^N), but it helps us narrow the playing field**

Our greatest hope is that we can get our models sufficiently refined so that we live in “flatland”

We don't have all the answers yet, but we have a path forward

- Continue to apply industry best practices and commercial experience to software quality
- Codify our assessments of numerical convergence
- Expand our understanding of the possible domains of our free parameters
- Expand sensitivity studies, screening efforts, and sampling techniques to span the hyperspace of input parameter combinations
- Develop new physics models that reduce our standard deviation for predicting multiple experiments

We will use margin to compensate for uncertainties as we make continuous progress