PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Dakota as a Solution Verification and Uncertainty Quantification Tool

Benjamin Pax
Purdue University

October 29, 2010

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

PURDUE UNIVERSITY

THE UNIVERSITY of NEW MEXICO

VANDERBILT UNIVERSITY

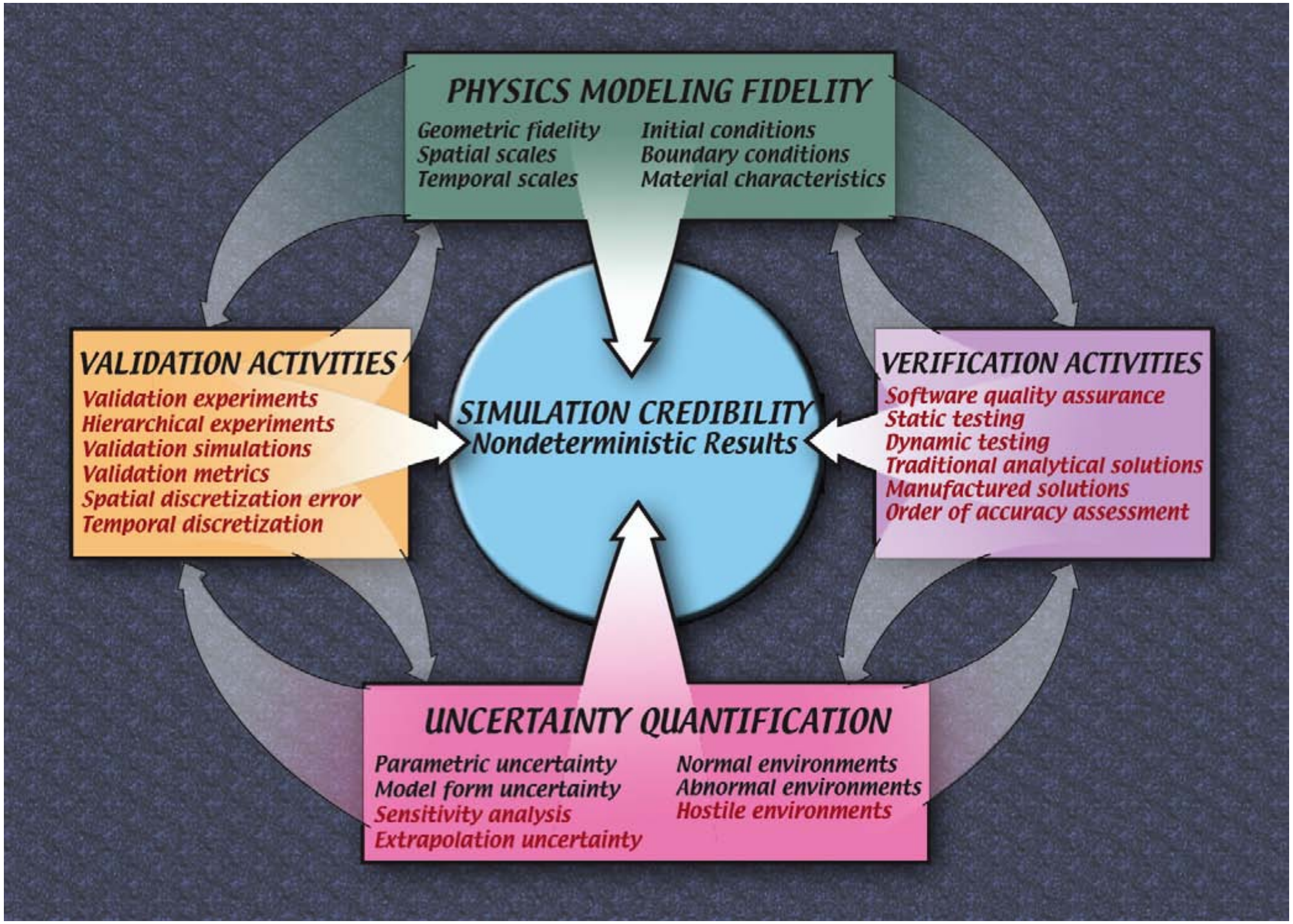
Outline

- Why perform Verification of your solution and What is Uncertainty Quantification?
- DAKOTA Overview
- Overview of Dakota Tools Using Analytical Beam Equation
  - Dakota Input Deck
  - Parameter Studies
  - Uncertainty Quantification
  - Mixed Aleatoric/Epistemic Uncertainty
- Case Study: Electrostatic Activation of Thin Film MEMS Device
  - Solution Verification
  - Aleatoric Uncertainty

Why perform Solution V&V and UQ?

- Verification
  - Determine if you are solving what you believe you are solving
  - Quantify how well you are solving the problem
- Uncertainty Quantification
  - Epistemic
    - Lack of knowledge
    - Reducible
  - Aleatoric
    - Variability inherent in the environment or problem definition
    - Irreducible, can't be reduced by further knowledge

Overview of V&V and UQ

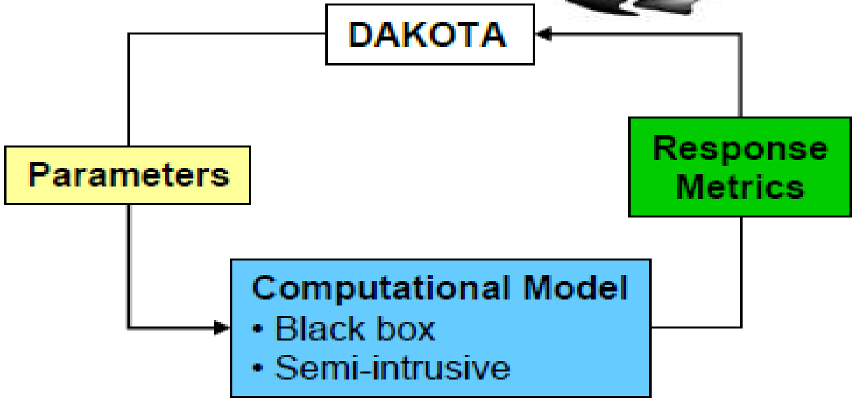


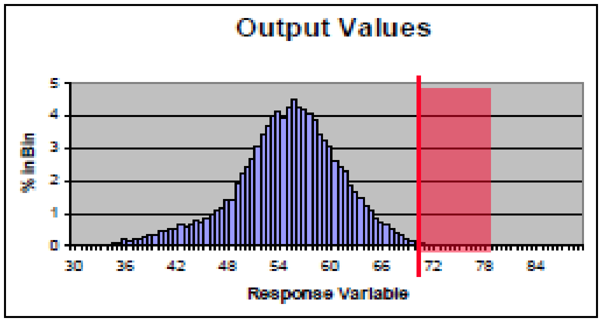
Oberkampf, W – courtesy Swiler, L.

Toolkit for Large-Scale Optimization & UQ

DAKOTA **allows analysis of fundamental science and engineering questions with computational models:**

- ***Sensitivities***: What are the crucial parameters?
- ***Uncertainties***: How safe, reliable, robust, variable is my system?
- ***Optimization***: What is the best performing design?
- ***Calibration***: What parameter values or models best match experimental data?





Example: Assessing probability of failure from distribution (uncertainty) of output values

DAKOTA analysis "strategies" rely on iterative analysis with a computational model for the phenomenon of interest

Strategies can be combined for more advanced capabilities, e.g.,

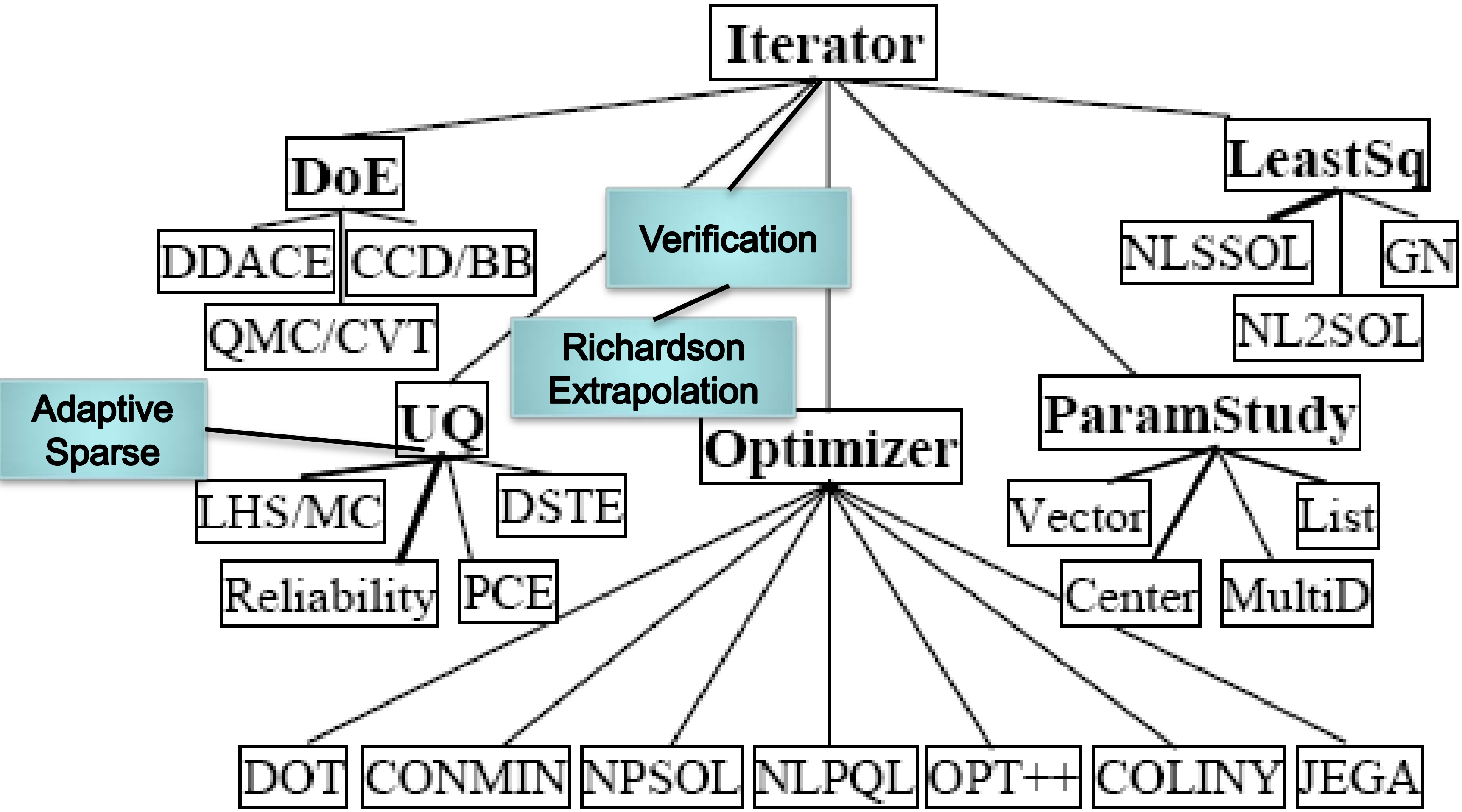
- ***Model calibration under uncertainty***
- ***Uncertainty of optima***

Multi-level parallelism supports large-scale applications and architectures

Broad deployment via open source model: Over 4,000 download registrations spanning government, industry, academia

Sandia National Laboratories

Dakota Features



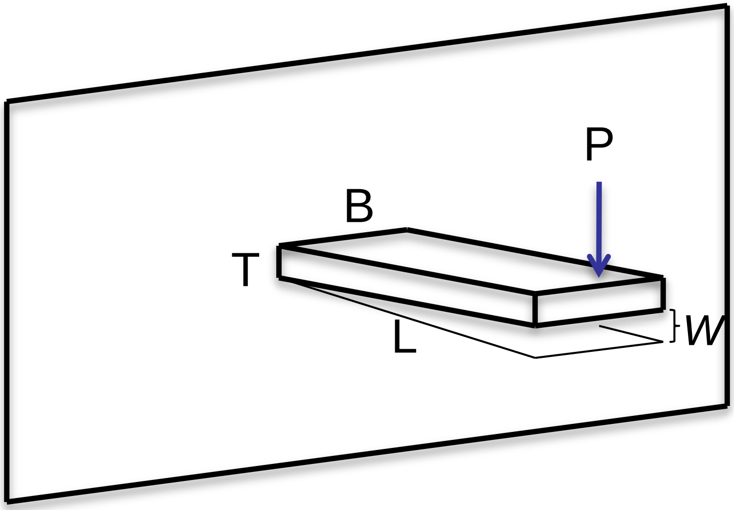
http://dakota.sandia.gov/software.html

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Dakota Input Deck

```
Strategy,
    single_method
    tabular_graphics_data
Method,
    multidim_parameter_study
        partitions = 3 3
Model,
    single
    primary_variable_mapping = ‘v1’ ‘v2’
    primary_response_mapping = 0. 1. 0. 1.
Variables,
    interval_uncertain
    num_intervals 1 1
    interval_probs 1. 1.
    interval_bounds 1. 10. 1. 10.
Responses,
    number_response_functions = 2
    response_descriptors ‘r1’ ‘r2’
Interface,
    asynchronous
    evaluation_concurrency = 3
    analysis_driver = ‘./simulation.py’
```

Example Problem: Analytical Beam



$$\frac{\partial^2}{\partial x^2}\left(EI\frac{\partial^2 w}{\partial x^2}\right) = q$$

$$I = \frac{BT^3}{12}$$

$$W = \frac{4PL^3}{EBT^3}$$

Parameters	UQ Type	Nominal Value	Std Dev Range	%
Youngs Modulus (E,Pa)	Epistemic	2.00E+11	2.00E+10	10.00%
Beam Width (b,m)	Aleatoric	0.15	0.015	10.00%
Beam Length (l,m)	Epistemic	0.5	0.05	10.00%
Beam Thickness (t,m)	Aleatoric	0.006	0.0006	10.00%
Load (P,N)	Epistemic	2000	200	10.00%
Density(rho,kg/m^3)	-	7874	-	-

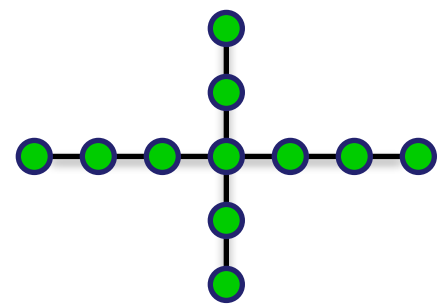
PRISM NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Example Problem: Parameter Study

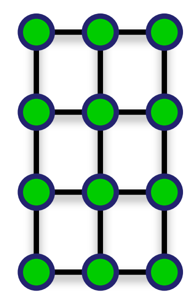
- List
  - Arbitrary set of points
    - Length of beam

- Vector
  - Ordered set of points in 1 dimension
    - Load on beam tip

- Centered
  - Evaluate 2 dimensions about 2 central values
    - Length of beam and Load on beam tip



- Multi-dimensional
  - Explore a full grid of points
    - Length of beam and Load on beam tip



Example Problem: Dakota Set-up

```
method,
    list_parameter_study
    list_of_points = 0.25 0.5 0.6 0.7 0.75 1.0

variables,
   discrete_state_set_real = 1
      set_values = 0.25 0.5 0.6 0.7 0.75 1.0
      descriptors       =  'L'
```

```
method,
    vector_parameter_study
    final_point = 4000
    num_steps = 9
variables,
    continuous_design = 1
       initial_point = 1000
       descriptors       =  'P'
```

```
method,
    centered_parameter_study
              step_vector = 0.05 200
              steps_per_variable = 2
variables,
    continuous_design = 2
              initial_point = 0.5 2000
              descriptors       =  'L' 'P'
```

```
method,
    multidim_parameter_study
    partitions = 2
variables,
    continuous_design = 2
       initial_point = 0.5 2000
       upper_bounds =  0.75 3000
       lower_bounds =  0.25 1000
       descriptors       = 'L' 'P'
```

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Example Problem: Dakota Set-up

```
method,
    nond_polynomial_chaos
    quadrature_order 4 4
        sample_type lhs
        seed = 1729 rng rnum2
        samples = 1000
variables,
    normal_uncertain = 2
        mean            =   0.15     0.006
        std_deviations  =   0.015    0.0006
        descriptors     =   'B'      'T'
```

```
method,
    nond_polynomial_chaos
    sparse_grid_level = 2
        sample_type lhs
        seed = 1729 rng rnum2
        samples = 1000
variables,
    normal_uncertain = 3
        mean            = 0.5    0.15    0.006
        std_deviations  = 0.05   0.015   0.0006
        descriptors     =  'L'    'B'     'T'
```

$$W = \frac{4PL^3}{EBT^3}$$

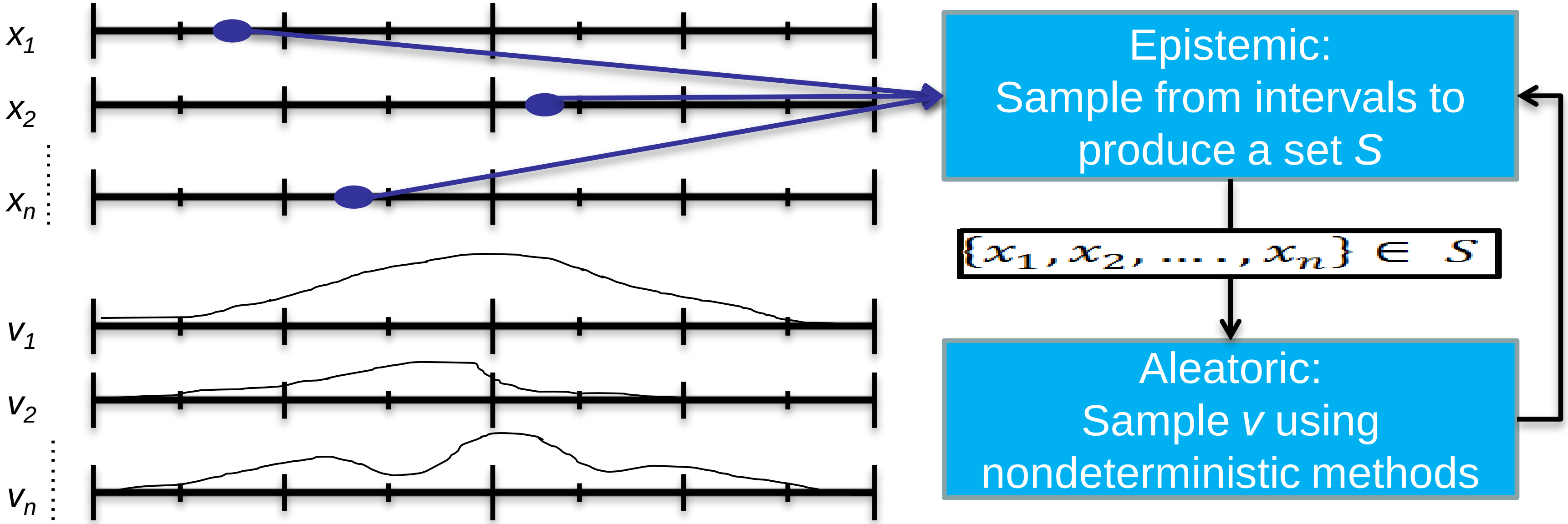
PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Example Problem: UQ Method Comparison

	Monte Carlo	LHC	Tensor	Sparse	Sparse 2
Simulations	6400	1280	16	5	17
Mean of W	1.668E-001	1.661E-001	1.660E-001	1.659E-001	1.660E-001
RMS error	-	4.24E-003	4.54E-003	5.35E-003	4.53E-003
% Error	-	0.4241%	0.4540%	**0.5347%**	0.4529%
Workload	-	20.000%	0.250%	**0.078%**	0.266%

Example Problem: Mixed Aleatoric and Epistemic Uncertainty

- Nested Model
  - Outer loop iterates over Epistemic
  - Inner loop iterates over Aleatoric
- Optimization, parameter studies, or sampling of Epistemic variables
- PCE, parameter studies, or sampling of Aleatoric variables



PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Example Problem: Dakota Set-up

```
strategy,
  single_method
  method_pointer = 'EPISTEMIC'
method,
      id_method = 'EPISTEMIC'
      model_pointer = 'EPIST_M'
      multidim_parameter_study
      partitions = 2 2 2
model,
      id_model = 'EPIST_M'
      nested
        variables_pointer  = 'EPIST_V'
        sub_method_pointer = 'ALEATORY'
        responses_pointer  = 'EPIST_R'
        primary_variable_mapping   = 'L' 'P' 'E'
        primary_response_mapping   = 1. 0. 0. 0. 0. 0.
                                     0. 1. 0. 0. 0. 0.
                                     0. 0. 0. 1. 0. 0.
                                     0. 0. 0. 0. 1. 0.
variables,
      id_variables = 'EPIST_V'
      interval_uncertain = 3
         num_intervals   = 1 1 1
         interval_probs  = 1.0 1.0 1.0
         interval_bounds = 0.4 0.6 1600 2400 1.8e11 2.2e11
responses,
      id_responses = 'EPIST_R'
      num_response_functions = 4
      response_descriptors = 'mean_W' 'STD_W' 'mean_Weight'  'STD_beta_Weight'
      no_gradients
      no_hessians
```

```
method,
      id_method = 'ALEATORY'
      model_pointer = 'ALEAT_M'
      nond_sampling sample_type lhs
      seed = 84952  samples = 5
        num_response_levels =  1 1
        response_levels = 0.2 3
        compute reliabilities
        complementary distribution
model,
      id_model = 'ALEAT_M'
      single
        variables_pointer = 'ALEAT_V'
        interface_pointer = 'ALEAT_I'
        responses_pointer = 'ALEAT_R'
variables,
      id_variables = 'ALEAT_V'
      continuous_design = 3
       normal_uncertain = 2

interface,
      id_interface = 'ALEAT_I'
      fork
       asynchronous evaluation_concurrency = 5
        analysis_driver = './list.py'
responses,
      id_responses = 'ALEAT_R'
      num_response_functions = 2
      response_descriptors = 'W' 'Weight'
      no_gradients
      no_hessians
```

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Case Study: Electrostatic Activation of Radio Frequency Capacitive Switch

- Coupled Physics
  - Electrostatics
  - Solid Mechanics
    - Contact
- Instability
- Loosely coupled
  - Iteration on non-linear loop
- Quasi-static

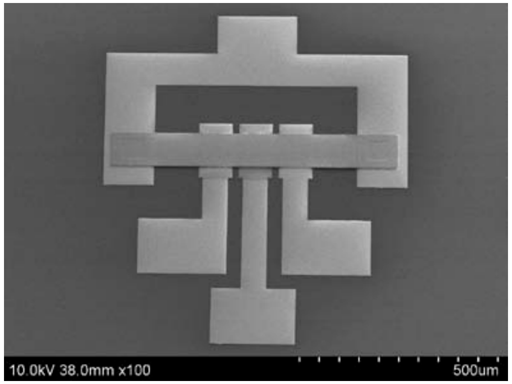
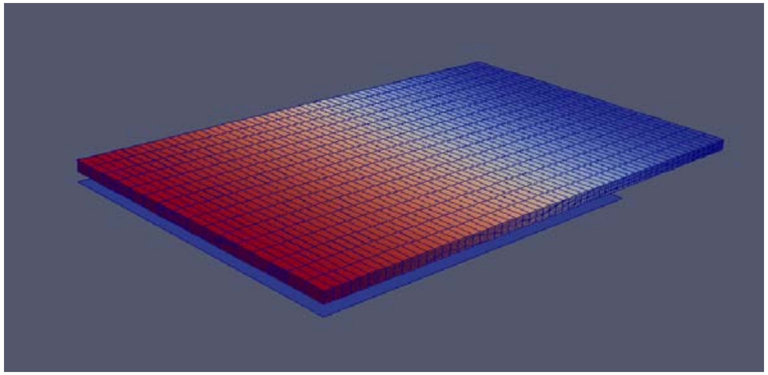


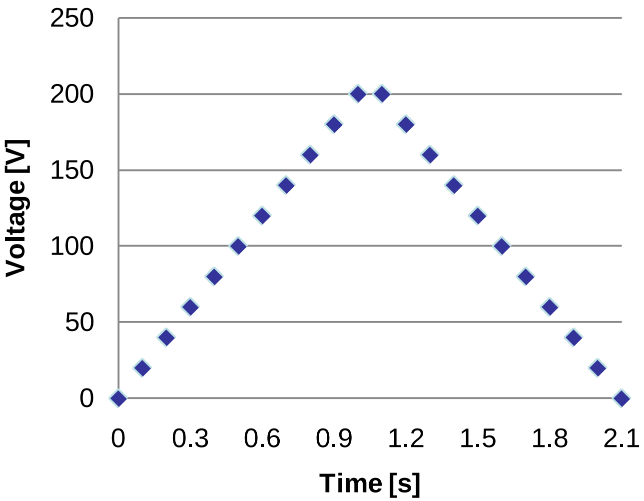
Image courtesy of NNSA sponsored Center for PRISM



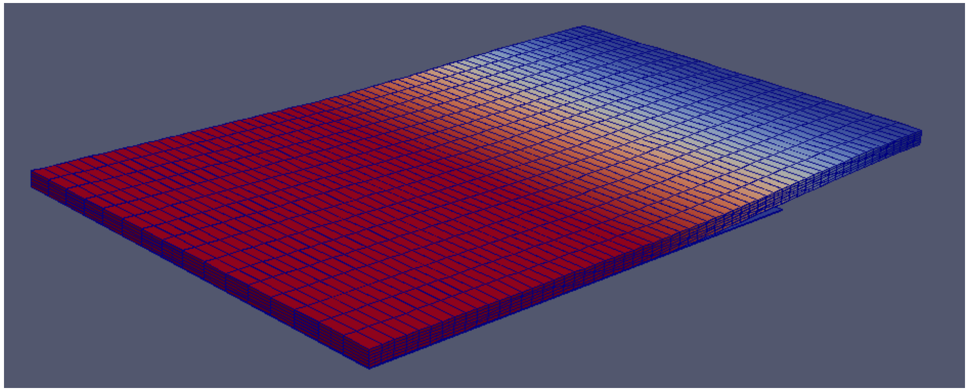
B. Pax 2010

Case Study: Beam Properties and Load Conditions

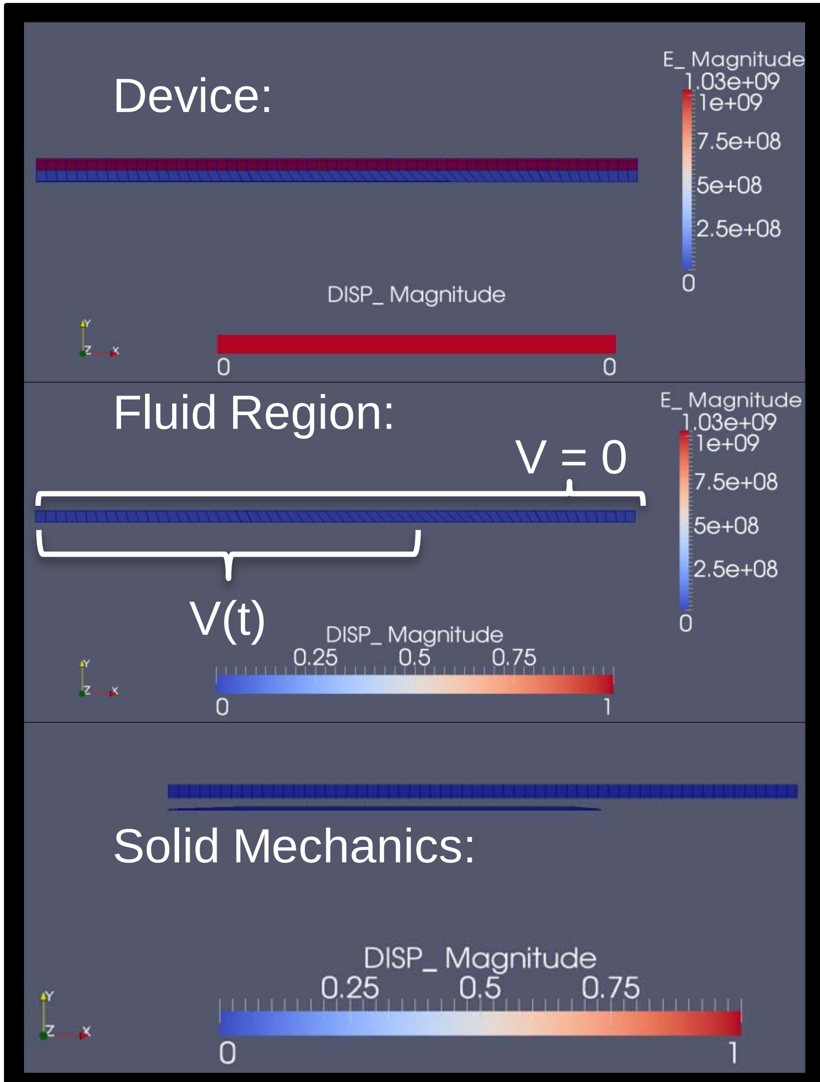
Voltage Load Boundary Condition

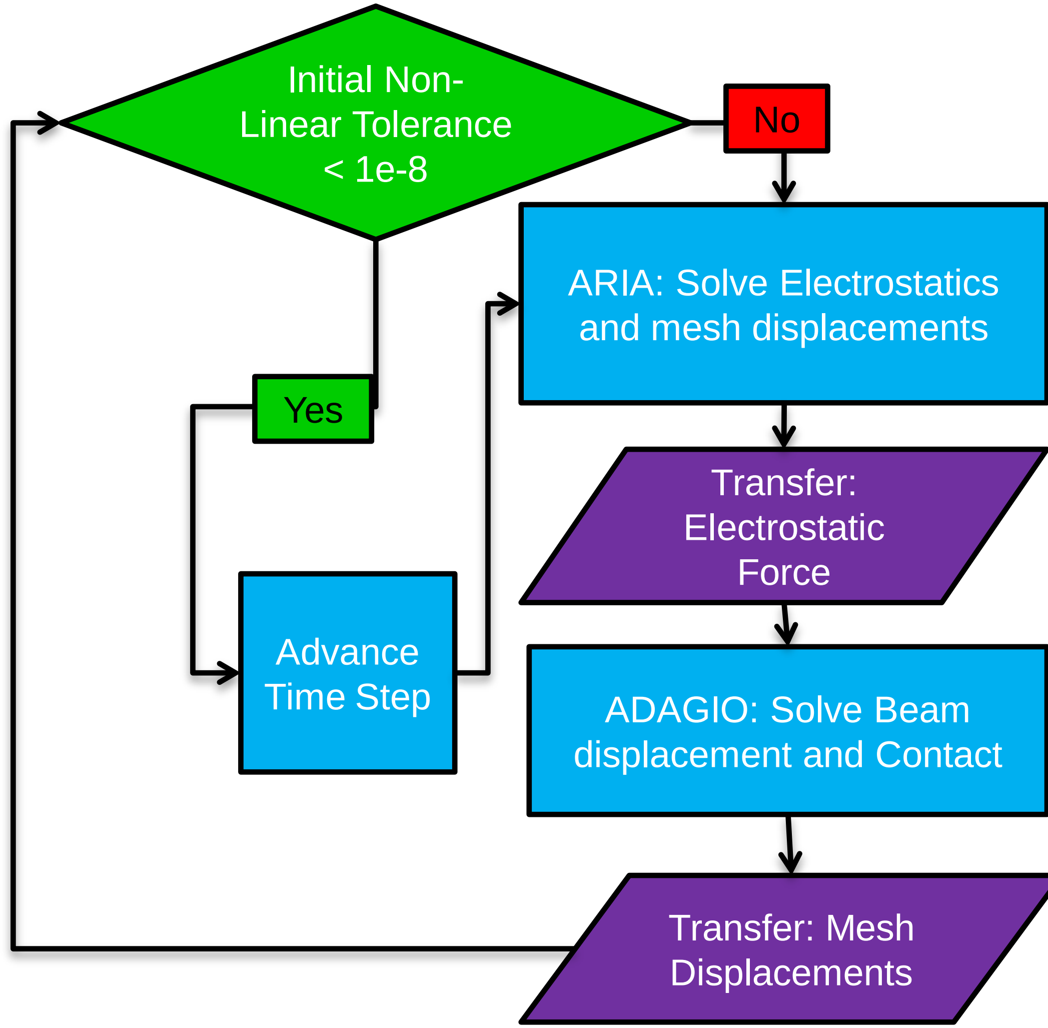


Parameters	UQ Type	Nominal Value	Std Dev Range	%
Young's Modulus (E) [Pa]	Epistemic	1.6E+11	5.00E+10	31.25%
Beam Width (T) [um]	Aleatoric	120	2	1.67%
Beam Length (Lb) [um]	Epistemic	400	2	0.50%
Electrode Length (Le) [um]	Aleatoric	270	1	0.37%
Beam Thickness (t) [um]	Aleatoric	4	0.6	15.00%
Gap Thickness (g) [um]	Aleatoric	3.8	0.3	7.89%
Dielectric Thickness (d) [um]	Aleatoric	0.15	0.02	13.33%
Density (rho) [kg/m^3]	-	8800	-	-



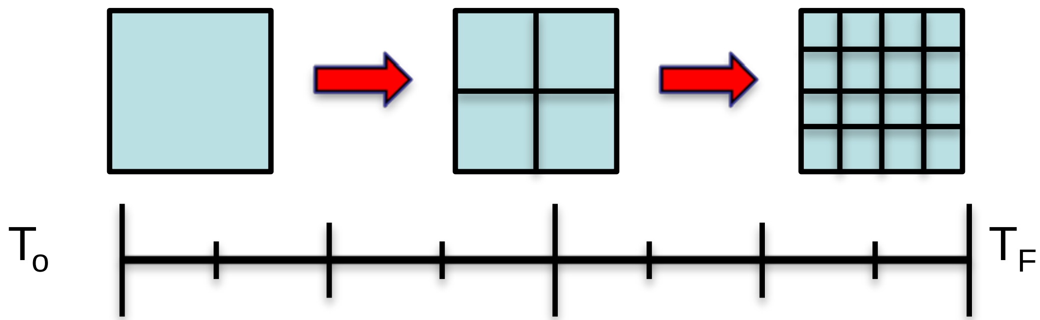
Case Study: Boundary Conditions and Solution Procedure





Case Study: Verification

- Code Verification (Algorithm)
  - Code solves equations correctly
    - Analytical Solutions
    - Method of Manufactured Solutions
- Solution Verification (Numeric's)
  - Simulations are refined to acceptable level of error
    - Discretization
    - Temporal
    - Tolerance



Quantify!

Case Study: Solution Verification Discretezation Set-up

```
method,
    list_parameter_study
    list_of_points = 1 2 3 4 5 6 7 8

variables,
   discrete_state_set_real = 1
     set_values = 1 2 3 4 5 6 7 8
     descriptors       =  'Mesh_multiplier'
```

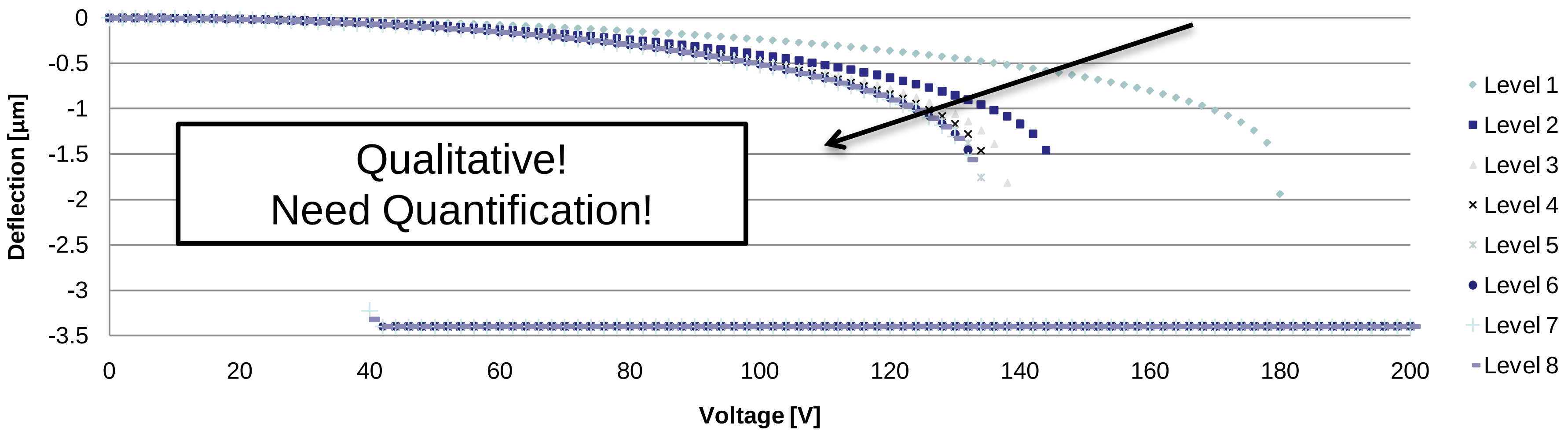
```
method,
    list_parameter_study
    list_of_points = 2 1 0.5 0.25 0.125

variables,
   discrete_state_set_real = 1
     set_values = 2 1 0.5 0.25 0.125
     descriptors       =  'V_step'
```

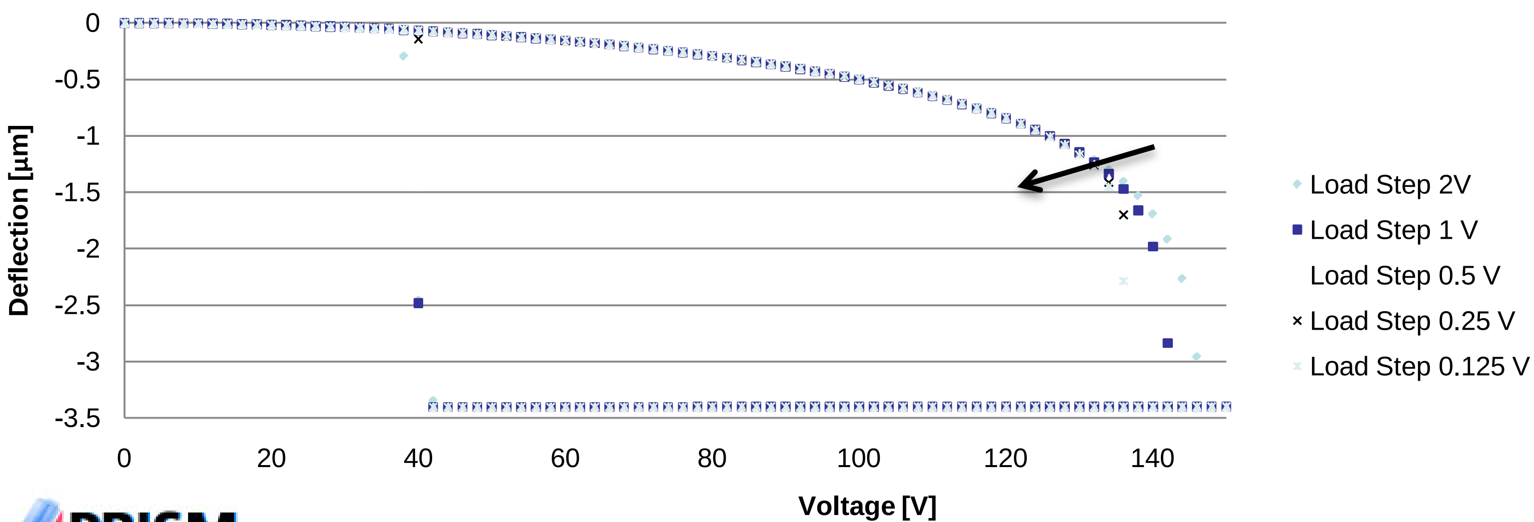
New Richardson Extrapolation Iterator!

```
method,
    richardson_extrap
    estimate_order
    refinement_rate = 2
variables,
   continuous_state = 1
     initial_state = 1
     descriptors       =  'Mesh_multiplier'
```

```
method,
    richardson_extrap
    estimate_order
    refinement_rate = 2
variables,
   continuous_state = 1
     initial_state = 1
     descriptors       =  '1/V_step'
```

Case Study: Solution Verification Discretization and Temporal





PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Case Study: Solution Verification - Convergence

$$p = \frac{\ln\left|\frac{\varepsilon_{32}}{\varepsilon_{21}}\right| + q(p^*)}{\ln|r_{21}|}$$

$$C = \frac{(\varphi_{fine} - \varphi_{medium})}{h^p_{medium} - h^p_{fine}}$$

$$\varphi_{extrap,coarse} = \varphi_{coarse} + C * h^p_{coarse}$$

$$r = \frac{h_{coarse}}{h_{fine}}$$

$$\varepsilon_{21} = \varphi_2 - \varphi_1$$

$$s = \sin\left(\frac{\varepsilon_{32}}{\varepsilon_{21}}\right)$$

$$q(p^*) = \ln\left(\frac{r_{21}^{p^*} - s}{r_{32}^{p^*} - s}\right)$$

Uniform refinement ratio:

$$q(p^*) = 0$$

ASME V&V 20-2009: Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Case Study: Solution Verification Discretization

Delta [V]	Mesh Factor	Pull-in	Rate	Extrapolated	Numerical Error [V]	% Error of Extrapolated
0.1	1	180.4	2.24	142.68	37.72	26.43%
0.1	2	145.9	2.45	136.97	8.93	6.52%
0.1	3	138.6	2.82	135.06	3.54	2.62%
0.1	4	135.9	2.42	133.98	1.92	1.43%
0.1	5	134.7	3.07	133.61	1.09	0.81%
0.1	6	134.0	1.87	133.04	0.96	0.72%
0.1	7	133.6	-	-		
0.1	8	133.3	-	-		

Discretization Error!
Can be used to bound results.

Delta [V]	Mesh Factor	Pull-in	Rate	Extrapolated	Num Error [V]	Error [%]
2	4	148	0.74	140.19	7.81	5.57%
1	4	143	0.42	136.14	6.86	5.04%
0.5	4	140	0.58	135.95	4.05	2.98%
0.25	4	137.75	0.78	135.48	2.27	1.68%
0.125	4	136.25	-	-	-	-
0.0625	4	135.375	-	-	-	-

Case Study: Solution Verification

- Large Error (5.6%, 7.81 V) for Pull-in voltage
- Dependence on load step size
  - Quasi-static problem
    - Convergence issue!
- Options to proceed
  - Improve simulation (Best Option)
    - Refine problem statement
    - Work with Code developers to improve capabilities, resolve convergence issues
  - Simulate using fine discretization/load step (Duct Tape)
    - Document shortcomings of simulation
      - Quantify!
    - Minimize effects of shortcomings

Case Study: Uncertainty Quantification Aleatoric Uncertainty

- Level 1 – Sparse Grid Integration of Polynomial Chaos Expansion
  - Effect Screening
- 6 uncertain parameters
  - Assume Normal distribution
    - Assumed form of distribution can be treated as an epistemic error!

Parameters	UQ Type	Nominal Value	Std Dev Range	%
Beam Width (T) [um]	Aleatoric	120	2	1.67%
Beam Length (Lb) [um]	Epistemic	400	2	0.50%
Electrode Length (Le) [um]	Aleatoric	270	1	0.37%
Beam Thickness (t) [um]	Aleatoric	4	0.6	15.00%
Gap Thickness (g) [um]	Aleatoric	3.8	0.3	7.89%
Dielectric Thickness (d) [um]	Aleatoric	0.15	0.02	13.33%

Case Study: Uncertainty Quantification Aleatoric Uncertainty – Results

Statistics derived analytically from polynomial expansion:

Moments for each response function:
Pull_in: Mean = 1.4941666667e+02 Std. Dev. = 3.5132665237e+01 Coeff. of Variation = 2.3513217114e-01
Pull_out: Mean = 3.5666666667e+01 Std. Dev. = 7.5346975166e+00 Coeff. of Variation = 2.1125320140e-01

Pull_in total Sobol indices:
7.8748822717e-03 Lb
2.4305192197e-05 Le
7.1209419612e-01 t
2.8000661641e-01 g
0.0000000000e+00 d
0.0000000000e+00 T

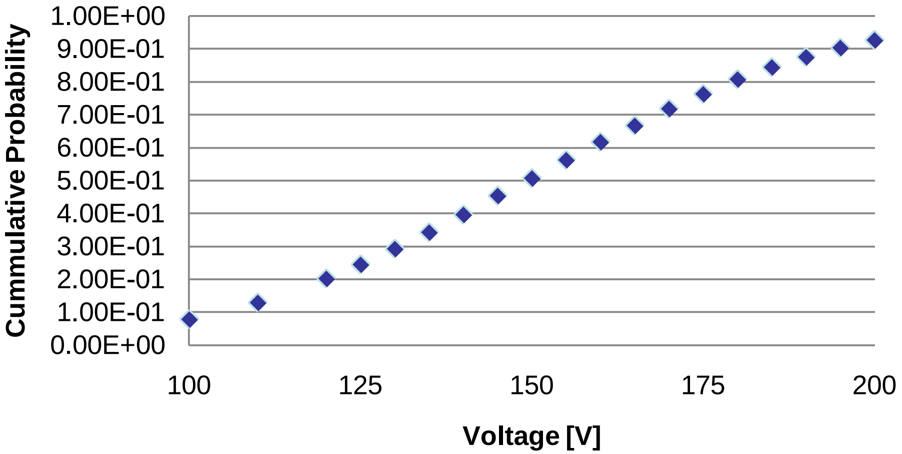
Pull_out total Sobol indices:
1.3210815254e-02 Lb
3.6696709039e-04 Le
6.0489387312e-01 t
1.9076417227e-01 g
1.9076417227e-01 d
0.0000000000e+00 T

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

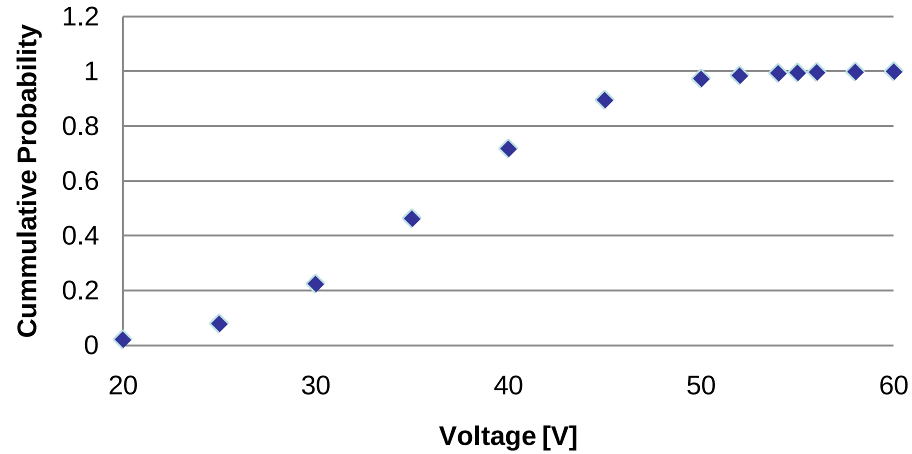
Case Study: Uncertainty Quantification Aleatoric Uncertainty - Results

```
Polynomial Chaos coefficients for Pull_in:
       coefficient   u1   u2   u3   u4   u5   u6
       ----------- ---- ---- ---- ---- ---- ----
   1.4941666667e+02  He0  He0  He0  He0  He0  He0
  -3.1176914536e+00  He1  He0  He0  He0  He0  He0
   1.7320508076e-01  He0  He1  He0  He0  He0  He0
   2.9646936323e+01  He0  He0  He1  He0  He0  He0
   1.8590678668e+01  He0  He0  He0  He1  He0  He0
   0.0000000000e+00  He0  He0  He0  He0  He1  He0
   0.0000000000e+00  He0  He0  He0  He0  He0  He1

Polynomial Chaos coefficients for Pull_out:
       coefficient   u1   u2   u3   u4   u5   u6
       ----------- ---- ---- ---- ---- ---- ----
   3.5666666667e+01  He0  He0  He0  He0  He0  He0
  -8.6602540378e-01  He1  He0  He0  He0  He0  He0
   1.4433756730e-01  He0  He1  He0  He0  He0  He0
   5.8601052323e+00  He0  He0  He1  He0  He0  He0
   3.2908965344e+00  He0  He0  He0  He1  He0  He0
   3.2908965344e+00  He0  He0  He0  He0  He1  He0
   0.0000000000e+00  He0  He0  He0  He0  He0  He1
---------------------------------------------------------------
```

CDF: Pull-in



CDF: Pull-out



PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Dakota: V&V and UQ tool

- Demonstrated solution verification using parameter studies
- Determined parameter relevance in uncertainty quantification using Sparse Grid PCE
- Evaluated PCE Response metrics to determine relative importance of parameters
  - Use to determine further course of action

http://dakota.Sandia.gov

Special Thanks

- Brian Carnes, Sandia V&V and UQ
- Brian Adams, Laura Swiler, Mike Eldred, Sandia Dakota Development Group
- Professor Murthy and Professor Alexeenko, Purdue Center for PRISM