SAND2016-9798C

Exceptional service in the national interest

Sandia National Laboratories

Gradient-Enhanced Polynomial Chaos Methods for Circuit Simulation

Eric Keiter, Laura Swiler, and Ian Wilcox
Sandia National Laboratories

October 7, 2016

11th International Conference on

Scientific Computing in Electrical Engineering

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Outline

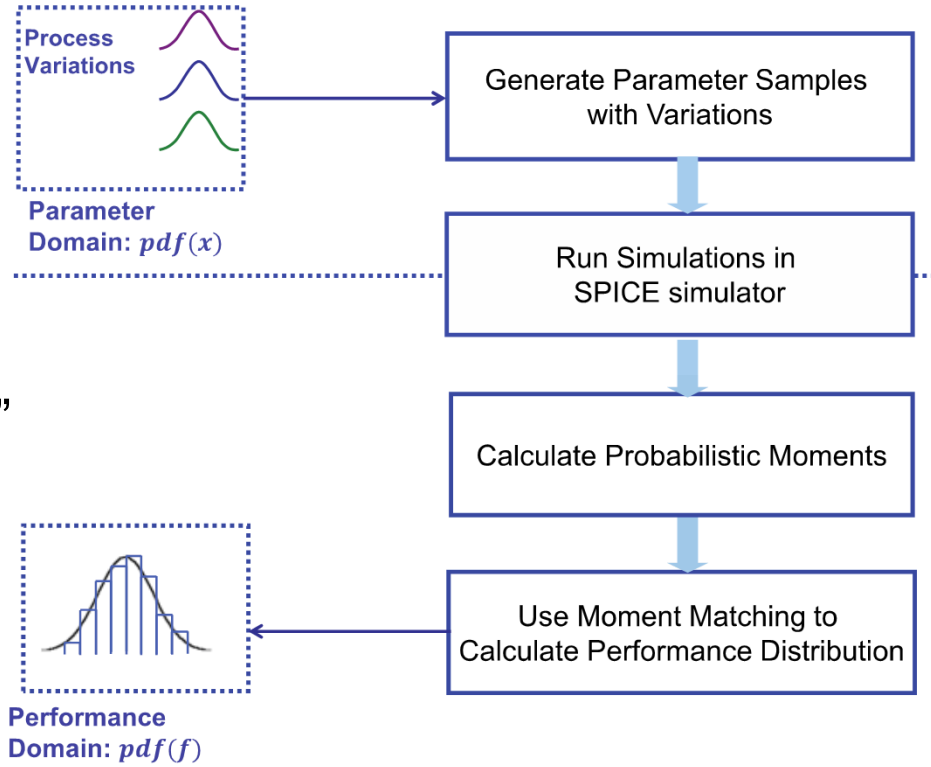
Sandia National Laboratories

- Xyce Overview
- Dakota Overview
- Adjoint capabilities in Xyce, both steady-state and transient
- Polynomial chaos overview
  - Gradient-Enhanced Polynomial chaos
- Gradient-Enhanced Polynomial chaos examples
  - Inverter chain circuit
  - Common emitter amplifier
  - Mixer circuit

Uncertainty Quantification (UQ)

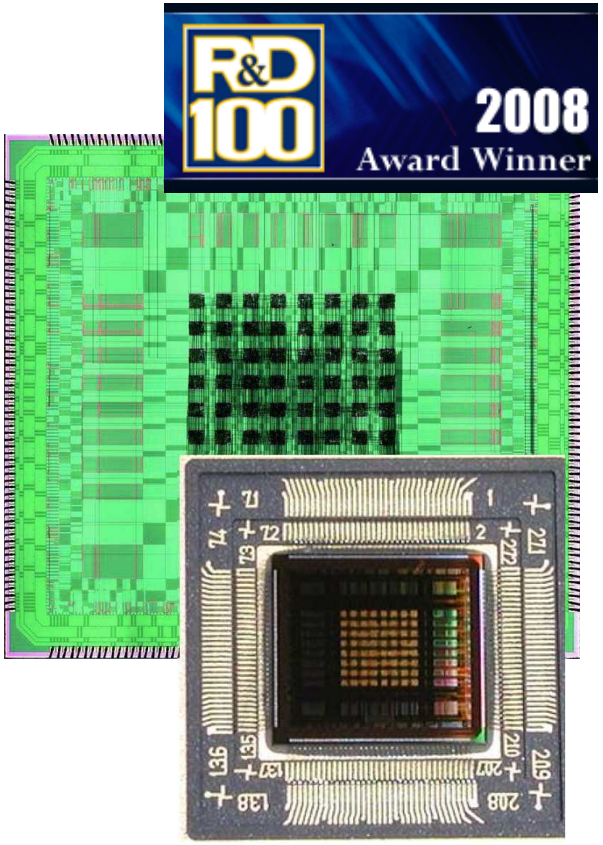
Sandia National Laboratories

- Uncertainty quantification (UQ) an important area
- General idea: given uncertainty on parameter inputs, how to estimate that on circuit outputs.
- Most simulators support “brute force” approaches based on sampling.
- Many much more sophisticated methods exist, including stochastic collocation methods, polynomial chaos, etc.



Parallel Circuit Simulator

- Xyce: Massively Parallel circuit simulator:
  - Distributed Memory Parallel (MPI-based)
  - Unique solver algorithms
  - SPICE-Compatible
  - Industry standard models (BSIM, PSP, EKV, VBIC, etc)
- Analysis types
  - DC, TRAN, AC
  - Harmonic Balance (HB)
  - Multi-time PDE (MPDE)
  - Model order reduction (MOR)
  - Direct and Adjoint sensitivity analysis
  - Uncertainty quantification (UQ) via Dakota.
- Sandia-specific models
  - Prompt Photocurrent
  - Prompt Neutron
  - Thermal
- Other, non-traditional models
  - Neuron/synapse
  - Reaction network
  - TCAD (PDE-based)
- Xyce Release 6.5 in May/June 2016
  - Open Source!
  - GPL v3 license



http://xyce.sandia.gov

Dakota enhances simulations…

Sandia National Laboratories

Algorithms for design exploration and simulation credibi

dakota.sandia.gov

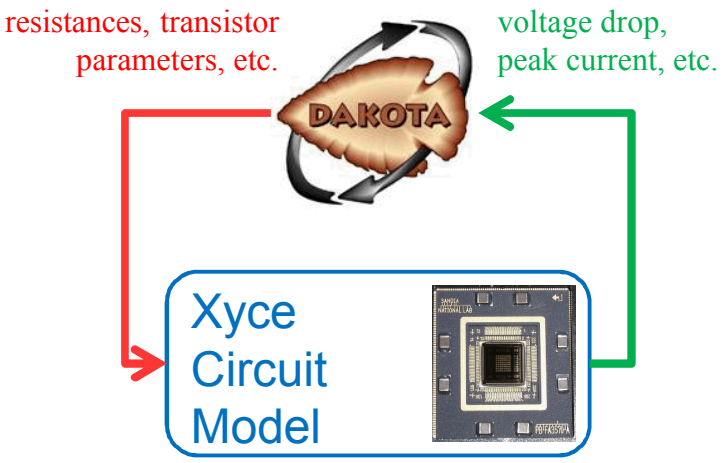
- Suite of iterative mathematical and statistical methods that interface to computational models
- Makes sophisticated parametric exploration of simulations practical for a computational design-analyze-test cycle

- Provides scientists and engineers (analysts, designers, decision makers) greater perspective on model predictions:
  - *Enhances understanding of risk by quantifying margins/uncertainties*
  - *Improves products through simulation-based design, calibration*
  - *Assesses simulation credibility through verification and validation*

…by analyzing ensembles

- Strategically selects model parameters
- Manages concurrent simulations
- Analyzes responses (model outputs)
- ➢ Automates one-pass parameter variation/ analysis to advanced goal-oriented studies

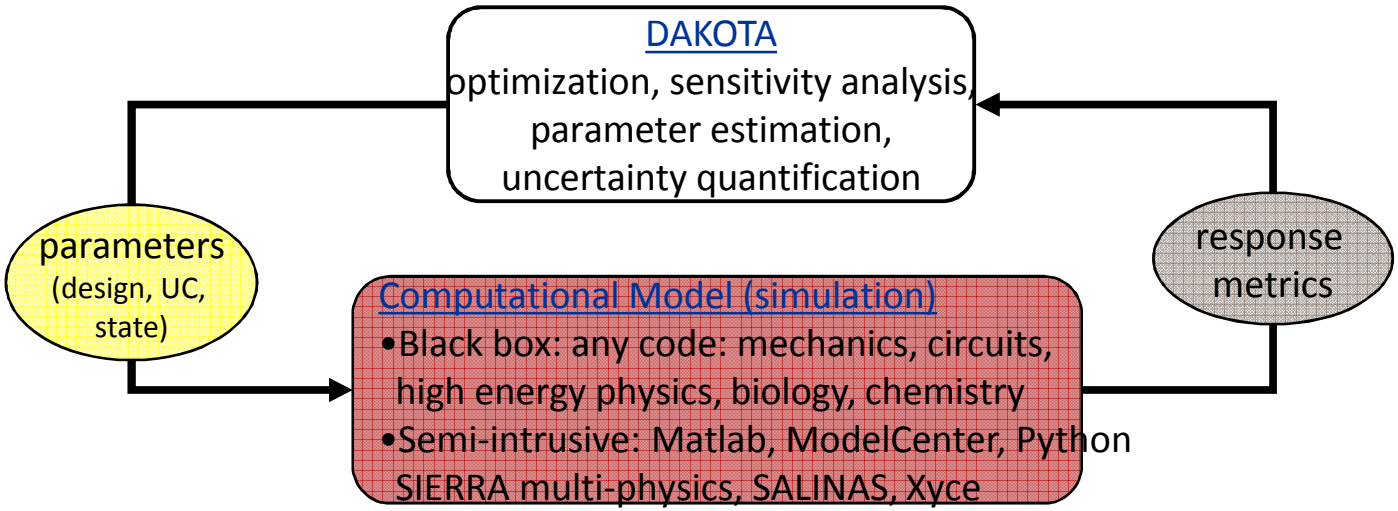
Run	Input	Output
1	0.814	91.3
2	0.906	63.24
…		
N	1.270	9.75



Automated Iterative Analysis of Computational Models

Sandia National Laboratories

Automate typical "parameter variation" studies with various advanced methods and an interface to your simulation



Transient Adjoint Sensitivities in Xyce

Eric Keiter, Sandia National Laboratories
11th International Conference on Scientific Computing in Electrical Engineering

Sandia National Laboratories

Why embedded parameter sensitivity are important

- Many advanced UQ techniques can use parameter derivatives.
  - Failure analysis methods
  - Regression-based Polynomial Chaos (PCE)
  - Gradient-based optimization

- Embedded sensitivities in an application code is better than relying on finite differences:
  - much faster (single solve)
  - much more accurate.

Parameter sensitivities

- Xyce (v6.5) now supports:
  - Direct and Adjoint steady state (.DC)
  - Direct and Adjoint transient (.TRAN)
- Parameter sensitivities (steady state):

$$\left(\frac{\partial F}{\partial x}\right)\frac{\partial x}{\partial p} = \frac{\partial F}{\partial p}$$

$$\frac{dO}{dp} = \frac{\partial O}{\partial x}\left(\frac{\partial F}{\partial x}\right)^{-1}\frac{\partial F}{\partial p}$$

Handled by the Xyce expression library

Xyce Jacobian Matrix (inverse)

Provided by device models (analytic or finite difference)

Direct

$$\frac{dO}{dp} = \frac{\partial O}{\partial x}\left[\left(\frac{\partial F}{\partial x}\right)^{-1}\frac{\partial F}{\partial p}\right]$$

Adjoint

$$\frac{dO}{dp} = \left[\frac{\partial O}{\partial x}\left(\frac{\partial F}{\partial x}\right)^{-1}\right]\frac{\partial F}{\partial p}$$

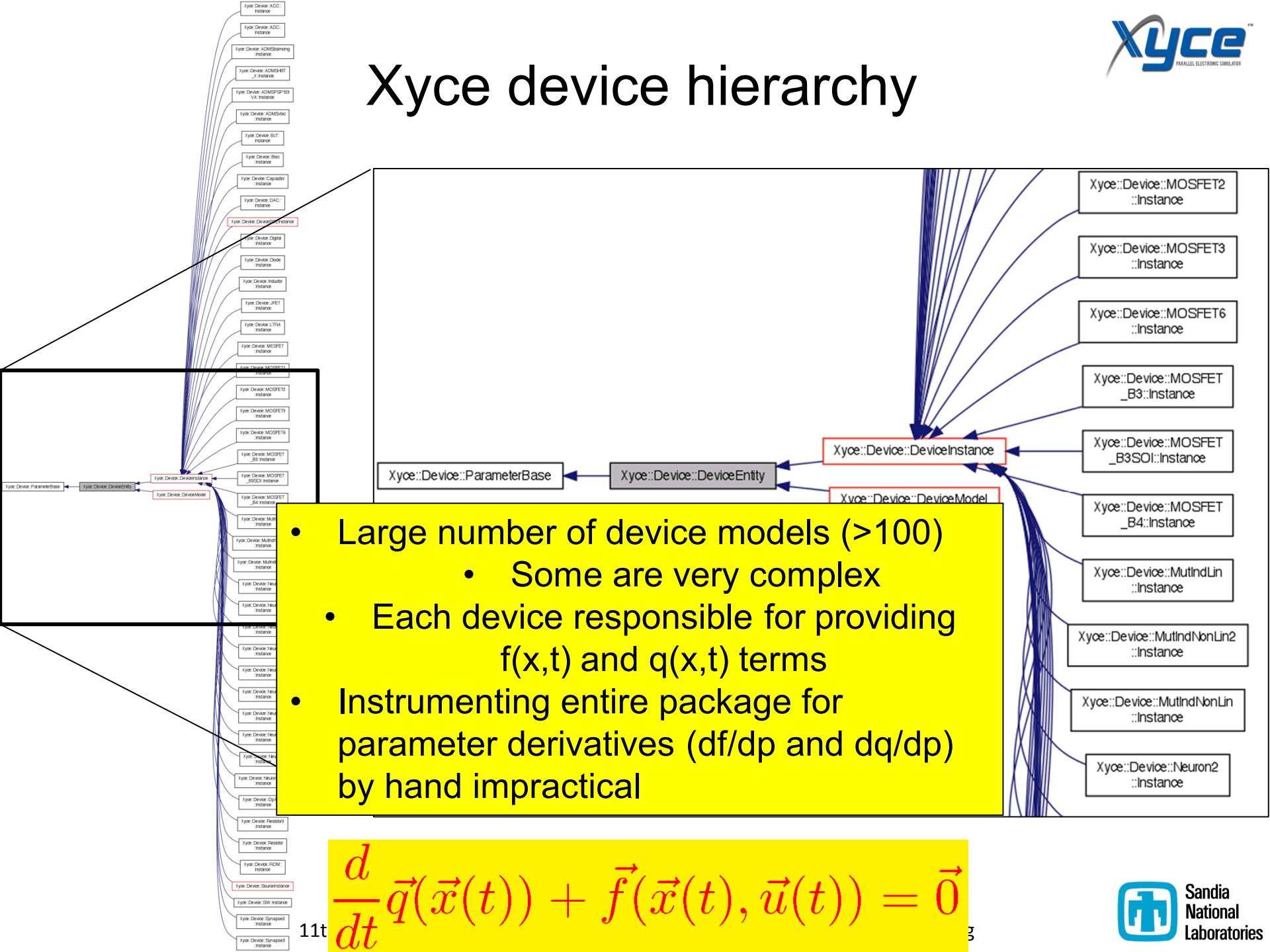
$$\left[\frac{\partial O}{\partial x}\left(\frac{\partial F}{\partial x}\right)^{-1}\right] = \left(\frac{\partial F}{\partial x}\right)^{-T}\frac{\partial O}{\partial x}$$

$$\left(\frac{\partial F}{\partial x}\right)^{T}\theta^* = \frac{\partial O}{\partial x}$$

$$\theta^* = \frac{\partial O}{\partial F}$$

O=objective function (scalar)
p=parameter (scalar)
F=residual vector
X=solution vector

Xyce device hierarchy



- Large number of device models (>100)
  - Some are very complex
- Each device responsible for providing f(x,t) and q(x,t) terms
- Instrumenting entire package for parameter derivatives (df/dp and dq/dp) by hand impractical

$$\frac{d}{dt}\vec{q}(\vec{x}(t)) + \vec{f}(\vec{x}(t), \vec{u}(t)) = \vec{0}$$

ADMS-Xyce

ADMS = *Automatic Device Model Synthesizer*

Verilog-A: industry standard format for new models

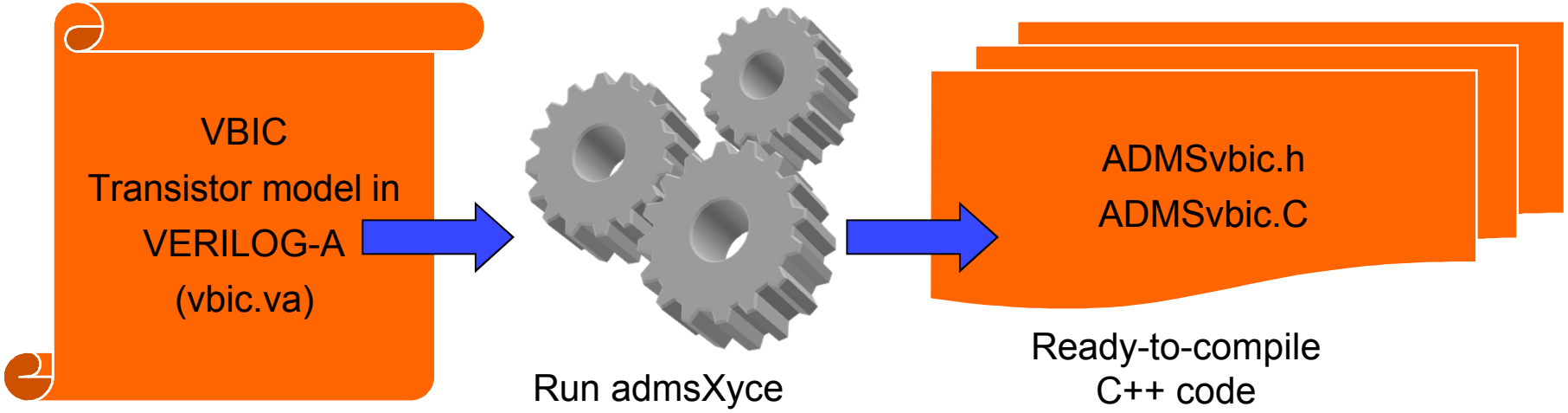
e.g. VBIC, Mextram, EKV, HiCUM, etc

ADMS translates Verilog-A to Xyce-compliant C/C++ code;

API automatically handles data structures, matrices, tedious details.

including analytic sensitivities via Sacado automatic differentiation

This capability enables practical application of advanced UQ methods



Computing Analytic Derivatives through Automatic Differentiation (AD)?

- Technique to compute analytic derivatives without hand-coding the derivative computation
- How does it work -- freshman calculus
  - Computations are composition of simple operations (+, *, sin(), etc…) with known derivatives
  - Derivatives computed line-by-line, combined via chain rule
- Derivatives accurate as original computation
  - No finite-difference truncation errors
- Provides analytic derivatives without the time and effort of hand-coding them

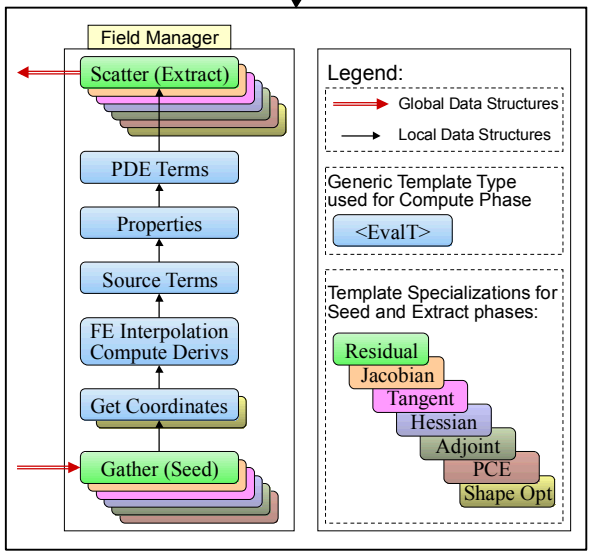
$$y = \sin(e^x + x \log x), \quad x = 2$$

	x	$\frac{d}{dx}$
$x \leftarrow 2$	2.000	1.000
$t \leftarrow e^x$	7.389	7.389
$u \leftarrow \log x$	0.301	0.500
$v \leftarrow xu$	0.602	1.301
$w \leftarrow t + v$	7.991	8.690
$y \leftarrow \sin w$	0.991	-1.188

Automatic Differentiation with Sacado

- Sacado: Trilinos AD tools for C++ applications
  - Templating and operator overloading
  - Robust, accurate, and efficient derivative evaluation
  - Jacobians, adjoints, Hessians, …
- Element-driven AD approach
  - Apply AD only at "element-fill" level
  - Developers write templated C++ code for element fill calculations (physics)
  - Handwritten glue code for initializing/extracting derivatives from/to global data structures
- Transformation to embedded algorithms
  - Supported by a number of Trilinos packages
  - Incorporated into a number of Sandia simulation tools
  - Enables R&D on numerous embedded algorithms

```
template <class ScalarType>
inline ScalarType simple_function(const ScalarType& u) {
  return 1.0/(std::pow(std::log(u),2.0) + 1.0);
}
```



Template-based generic programm
Pawlowski et al, MS 30

ADMS-Xyce

ADMS-converted models in Xyce (which now include analytic parameter sensitivities):

- Bipolar models:
  - VBIC
  - MEXTRAM
  - FBH HBT_X version 2.1
  - FBH HBT_X version 2.3
- CMOS models:
  - EKV
  - PSP (version 103)
  - BSIM-CMG 107
  - BSIM6
  - MVS Silicon virtual source model
- Radiation models:
  - HBT neutron models
- These models (with analytic derivatives) are all available as of Xyce 6.3.
- Impact: QASPR, W88ALT, Enhanced Surveillance, …

Steady-State Sensitivities

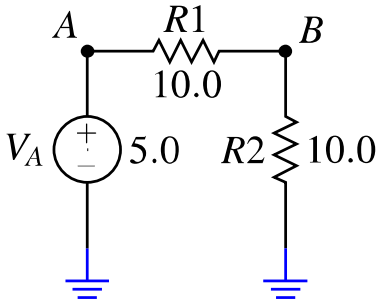
- Adjoint and direct DCOP sensitivities have been in Xyce for a long time.
- Solves for dO/dp, where O=objective function, p=parameter.
- Can be applied to *any* device model parameter.

- Simple voltage divider example.
- Objective function is:

$$O = \frac{1}{2}\left(V(B) - 3\right)^2$$

- Parameters are: R1 and R2
- The analytic solutions are:

$$\frac{dO}{dR1} = -\frac{dO}{dR2} = \frac{1}{16} = 6.25e-2$$



Xyce Input File

```
Voltage divier
R1 A B 10.0
R2 B 0 10.0
Va A 0 5

.dc Va 5 5 1
.print dc v(A) v(B)

.print sens
.SENS objfunc={0.5*(V(B)-3.0)**2.0} param=R1:R,R2:R
.options SENSITIVITY direct=1 adjoint=1

.END
```

Xyce Output

```
Direct Sensitivities of objective function:{0.5*(V(B)-3.0)**2.0}
Name         Value    Sensitivity    Normalized
R1:R    1.0000e+01     6.2500e-02     6.2500e-03
R2:R    1.0000e+01    -6.2500e-02    -6.2500e-03

Adjoint Sensitivities of objective function:{0.5*(V(B)-3.0)**2.0}
Name         Value    Sensitivity    Normalized
R1:R    1.0000e+01     6.2500e-02     6.2500e-03
R2:R    1.0000e+01    -6.2500e-02    -6.2500e-03
```

Transient Direct Sensitivities

- Available starting in Xyce v6.2 (fall, 2014)
- Solves for dO/dp, where O=objective function, p=parameter.
- Can be applied to *any* device model parameter.
- In general, direct form solves

$$\frac{dO}{dp} = \frac{dO}{dx}\left[\left(\frac{dF}{dx}\right)^{-1}\frac{dF}{dp}\right] + \frac{dO}{dp}$$

- For transient, the linear system being solved is specific to the integration method. For backward Euler, this gives:

$$\left[\frac{1}{h}\frac{dq}{dx} + \frac{dj}{dx}\right]\frac{dx}{dp}_{n+1} = -\frac{1}{h}\left[\frac{dq}{dp}_{n+1} - \frac{dq}{dp}_{n}\right] - \frac{dj}{dp} + \frac{db}{dp} + \frac{1}{h}\left[\frac{dq}{dx}\right]\frac{dx}{dp}_{n}$$

- The RHS dq/dp, dj/dp, db/dp terms come from **compact device models**.
- Xyce will compute these automatically; method depends on device implementation
  - Hand-coded derivatives in simple devices (resistors, etc)
  - Automatic-differentiation (Sacado) in devices that have been instrumented for it
  - Finite difference derivatives if neither are available.
- Transient Adjoint sensitivities planned for this summer/fall.

Transient Direct Sensitivities

- Example circuit.
- Simple diode, driven by sinewave source.
- Comparison to finite difference sensitivity.

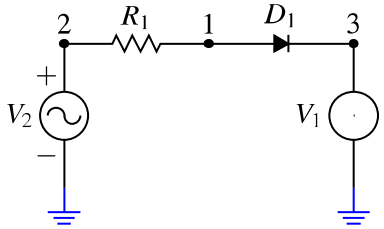


Figure 4.9. Transient Diode Circuit

Xyce Input File

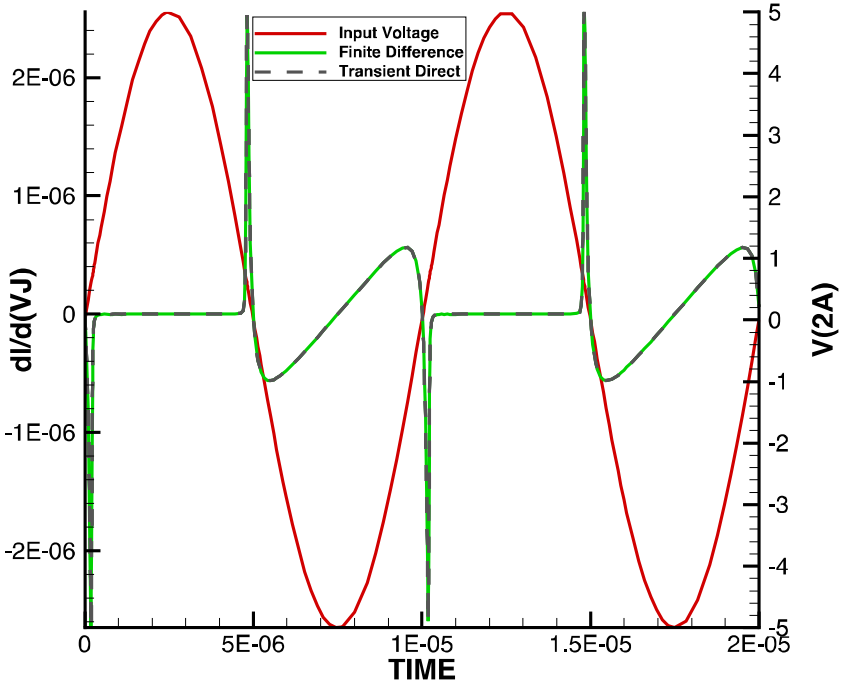
```
transient diode circuit sensitivity calculation
R 1 2 0.0001
V2 2 0 0.0 SIN(0 5 100K)
V1 3 0 0.0

D2 1 3 DZR
.MODEL DZR D( level=2 IS = 1E-14 RS = 10.8 N = 1 TT = 0
+ CJO = 1P VJ = 1 M = .5 EG = 1.11
+ XTI = 3 KF = 0 AF = 1 FC = .5
+ BV = 7.255 IBV = .001 tbv1 = 0.00013 tbv2 = -5e-8 )

.SENS objfunc={I(V1)}
+ param=DZR:VJ,DZR:CJO,DZR:EG,DZR:XTI,DZR:M,DZR:IS,DZR:RS,DZR:N

.options SENSITIVITY adjoint=0 direct=1

.options timeint method=gear
.TRAN 0 2e-5
.print TRAN format=tecplot v(2) I(V1)
.print SENS format=tecplot v(2)
.options device temp=25
.END
```



Transient Adjoint Sensitivities

- Available starting in Xyce v6.5 (Spring, 2016)
- Solves for dO/dp, where O=objective function, p=parameter.
- Can be applied to *any* device model parameter.
- In general, adjoint form solves

$$\frac{dO}{dp} = \left[\frac{\partial O}{\partial x}\left(\frac{\partial F}{\partial x}\right)^{-1}\right]\frac{\partial F}{\partial p}$$

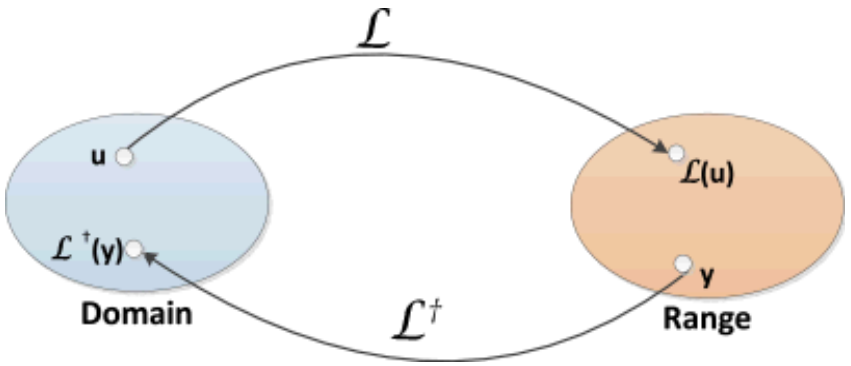
- Adjoint solutions are generally concerned with preserving an inner product, i.e. $\langle L\{u\}, y\rangle_R = \langle u, L^{\dagger}\{y\}\rangle_D$.

• The L operator represents the original differential equation:

$$\frac{d}{dt}C(t)\,x(t) + G(t)\,x(t) = u(t).$$

•For the transient case it can be shown that the adjoint, $z = L^{\dagger}\{y\}$ is the solution to the differential equation:

$$-C^{\dagger}(t)\frac{d}{dt}z(t) + G^{\dagger}(t)z(t) = y(t),$$



A. Meir and J. Roychowdhury, "BLAST: Efficient computation of nonlinear delay sensitivities in electronic and biological networks using barycentric lagrange enabled transient adjoint analysis," *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, San Francisco, CA, 2012, pp. 301-310. doi: 10.1145/2228360.2228417

Transient Adjoint Sensitivities

- To understand why a backward integration is necessary, consider the transient direct system as a large block system.
  - Each block represents a different time point.

Time points: T0 .... TN

- Forward (direct method) Jacobian matrix is lower triangular.
  - Solving requires starting at initial time T0 and integrating forward.

$$\frac{\partial \mathbf{\Psi}(\mathbf{X})}{\partial \mathbf{X}} = \begin{bmatrix} \left(\frac{\partial F_0}{\partial x_0}\right) & & & & & & \\ \left(\frac{\partial F_1}{\partial x_0}\right) & \left(\frac{\partial F_1}{\partial x_1}\right) & & & & & \\ \left(\frac{\partial F_2}{\partial x_0}\right) & \left(\frac{\partial F_2}{\partial x_1}\right) & \left(\frac{\partial F_2}{\partial x_2}\right) & & & & \\ \left(\frac{\partial F_3}{\partial x_0}\right) & \left(\frac{\partial F_3}{\partial x_1}\right) & \left(\frac{\partial F_3}{\partial x_2}\right) & \left(\frac{\partial F_3}{\partial x_3}\right) & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ & & & & & \left(\frac{\partial F_{N-1}}{\partial x_{N-1}}\right) & \\ & & & & & \left(\frac{\partial F_N}{\partial x_{N-1}}\right) & \left(\frac{\partial F_N}{\partial x_N}\right) \end{bmatrix}$$

$$\frac{\partial \mathbf{\Psi}}{\partial \mathbf{X}}\Theta = \frac{d\mathbf{\Psi}}{dp} \qquad \mathbf{\Theta} = \left[\frac{dx_0}{dp}, \frac{dx_1}{dp}, ..., \frac{dx_N}{dp}\right]$$

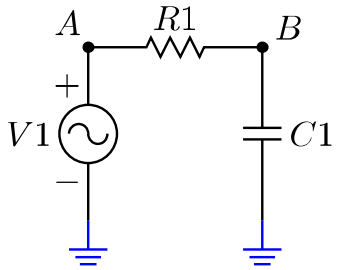
- Adjoint method Jacobian matrix (transpose of direct) is upper triangular.
  - To solve must backsolve starting from final time TN.

$$\left(\frac{\partial \mathbf{\Psi}(\mathbf{X})}{\partial \mathbf{X}}\right)^T = \begin{bmatrix} \left(\frac{\partial F_0}{\partial x_0}\right)^T & \left(\frac{\partial F_1}{\partial x_0}\right)^T & \left(\frac{\partial F_2}{\partial x_0}\right)^T & \left(\frac{\partial F_3}{\partial x_0}\right)^T & & & \\ & \left(\frac{\partial F_1}{\partial x_1}\right)^T & \left(\frac{\partial F_2}{\partial x_1}\right)^T & \left(\frac{\partial F_3}{\partial x_1}\right)^T & & & \\ & & \left(\frac{\partial F_2}{\partial x_2}\right)^T & \left(\frac{\partial F_3}{\partial x_2}\right)^T & & & \\ & & & \left(\frac{\partial F_3}{\partial x_3}\right)^T & & & \\ & & & & \ddots & \vdots & \vdots \\ & & & & & \left(\frac{\partial F_{N-1}}{\partial x_{N-1}}\right)^T & \left(\frac{\partial F_N}{\partial x_{N-1}}\right)^T \\ & & & & & & \left(\frac{\partial F_N}{\partial x_N}\right)^T \end{bmatrix}$$

$$\left(\frac{\partial \mathbf{\Psi}}{\partial \mathbf{X}}\right)^T \Theta^\star = \frac{dO}{d\mathbf{X}} \qquad \mathbf{\Theta}^\star = \left[\frac{dO}{dF_0}, \frac{dO}{dF_1}, ..., \frac{dO}{dF_N}\right]$$

Transient Adjoint Sensitivity Result

- Result for a simple RC circuit.



Xyce Input File

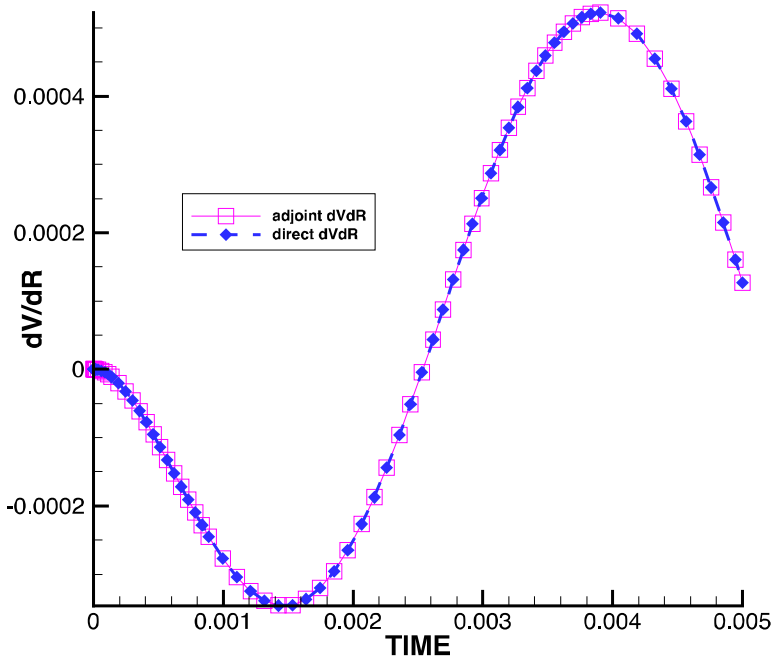
```
Test to show transient adjoint sensitivities for order-2 Gear integration.
*********************************************************************************
.param cap=1e-6
.param res=1e3

V1 1 0 0.0 sin (0.0 1.0 200 0.0 0.0 0.0 )
R1 1 2 {res}
C1 2 0 {cap}

.tran 1.0e-6 0.5e-2
.print tran format=tecplot V(1) V(2)
.print sens format=tecplot V(1) V(2)

.options timeint method=gear
.SENS objfunc={V(2)} param=R1:R,C1:C
.options SENSITIVITY direct=0 adjoint=1  OUTPUTLAMBDA=1
+ OUTPUTTRANSIENTADJOINT=1  FULLADJOINTTIMERANGE=1

.end
```



Transient Adjoint Sensitivities

- Example circuit.
- Simple diode, driven by sinewave source.
- Comparison to transient direct sensitivity.

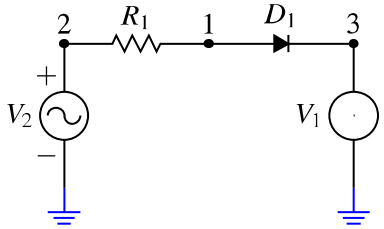


Figure 4.9. Transient Diode Circuit

Xyce Input File

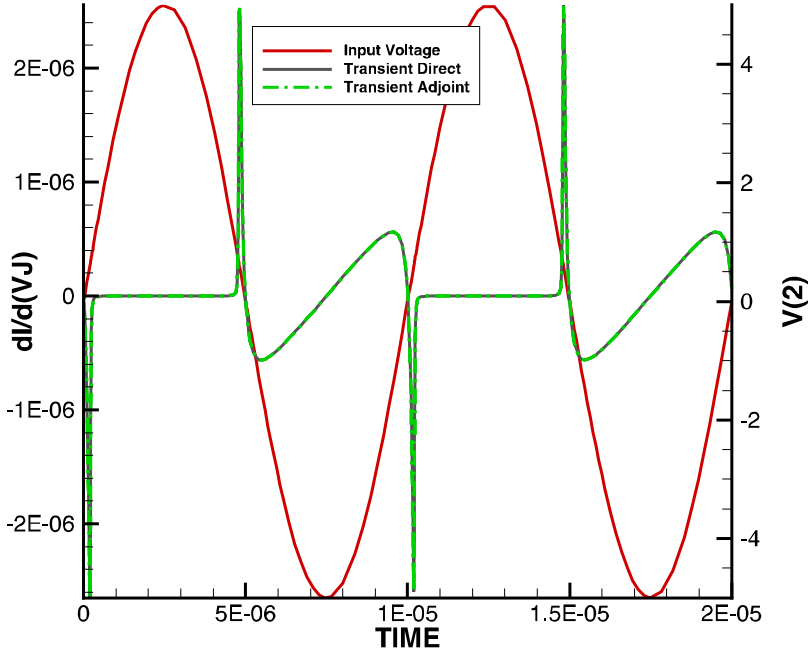
```
transient diode circuit sensitivity calculation
R 1 2 0.0001
V2 2 0 0.0 SIN(0 5 100K)
V1 3 0 0.0

D2 1 3 DZR
.MODEL DZR D( level=2 IS = 1E-14 RS = 10.8 N = 1 TT = 0
+ CJO = 1P VJ = 1 M = .5 EG = 1.11
+ XTI = 3 KF = 0 AF = 1 FC = .5
+ BV = 7.255 IBV = .001 tbv1 = 0.00013 tbv2 = -5e-8 )

.SENS objfunc={I(V1)}
+ param=DZR:VJ,DZR:CJO,DZR:EG,DZR:XTI,DZR:M,DZR:IS,DZR:RS,DZR:N

.options SENSITIVITY adjoint=1 direct=0

.options timeint method=gear
.TRAN 0 2e-5
.print TRAN format=tecplot v(2) I(V1)
.print SENS format=tecplot v(2)
.options device temp=25
.END
```

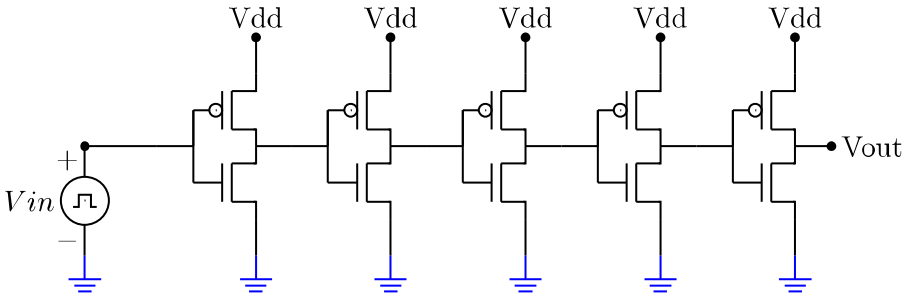


Xyce Transient Adjoint vs
Xyce Transient Direct Sensitivities

Sandia National Laboratories

Transient Sensitivities Example
5-stage CMOS inverter

- BSIM6 Model
- 5 stages.
- Each stage adds to signal delay

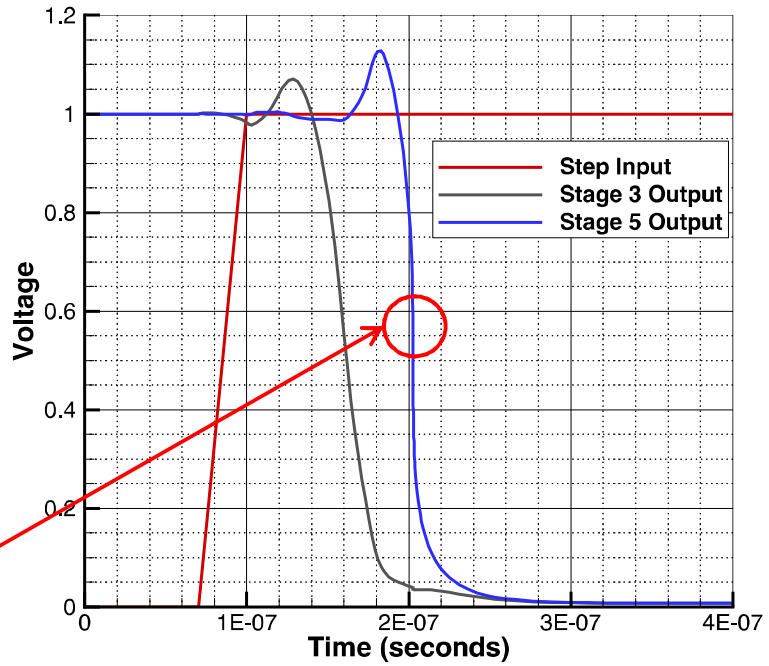


- Capacitive effects are the main delay contributors
  - PMOS and NMOS oxide thickness are the design parameters.
- Step input (Vin),Output at Vout.
- Objective function: Generalized Elmore delay.

$$O = \texttt{Elmore Delay} = \frac{\int_0^T g'_A(t) \cdot t \cdot dt}{\int_0^T g'_A(t) dt}$$

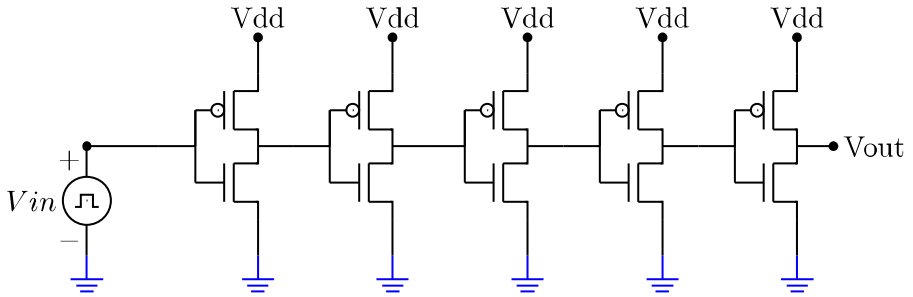
$$g_A(t) = \texttt{Vout}$$

- Elmore delay should give the approximate time for signal rise.



reduced objective (Elmore) = 2.078232472421e-007
reduced objective sensitivity for param dV(VO)d(nmos:TOXE)_Dir = 3.580676543124e+001
reduced objective sensitivity for param dV(VO)d(pmos:TOXE)_Dir = 1.962028129681e+001

Transient Sensitivities Example
5-stage CMOS inverter

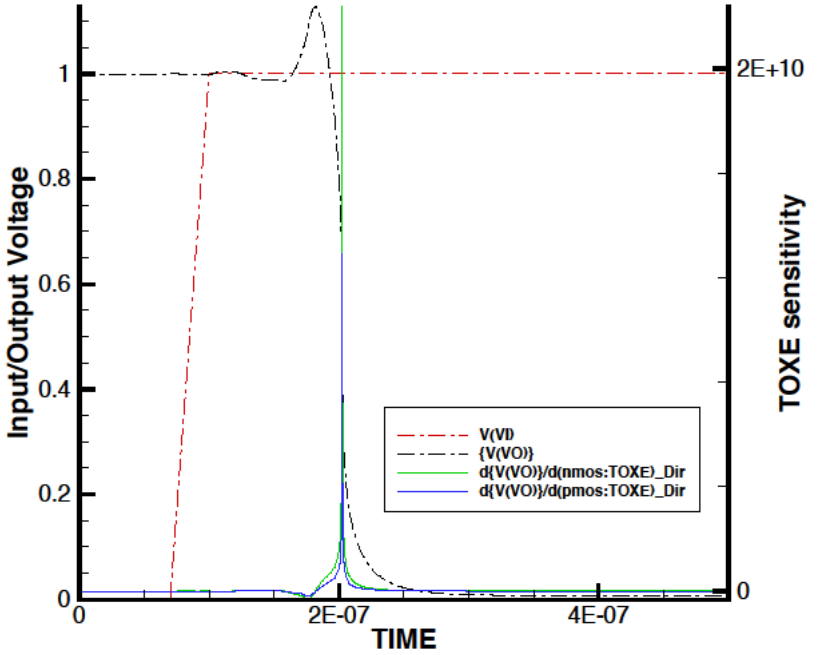


- BSIM6 Model
- 5 stages.
- Each stage adds to signal delay
- Capacitive effects are the main delay contributors
  - PMOS and NMOS oxide thickness are the design parameters.
- Step input (Vin),Output at Vout.
- Objective function: Generalized Elmore delay.

$$O = \texttt{Elmore Delay} = \frac{\int_0^T g'_A(t) \cdot t \cdot dt}{\int_0^T g'_A(t) dt}$$

$$g_A(t) = \texttt{Vout}$$

- Elmore delay should give the approximate time for signal rise.



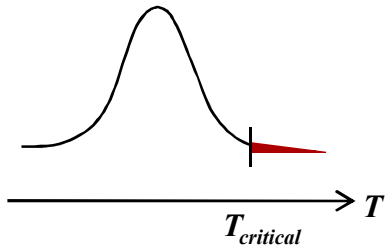
Gradient-Enhanced Uncertainty Quantification (UQ)

Eric Keiter, Sandia National Laboratories
11th International Conference on Scientific Computing in Electrical Engineering

Challenge: Calculating Potentially Small Probability of Failure

Sandia National Laboratories

- Given uncertainty in materials, geometry, and environment, how to determine likelihood of failure: *Probability*($T \geq T_{critical}$)?
- Perform 10,000 LHS samples and count how many exceed threshold;
  (better) perform a reliability method



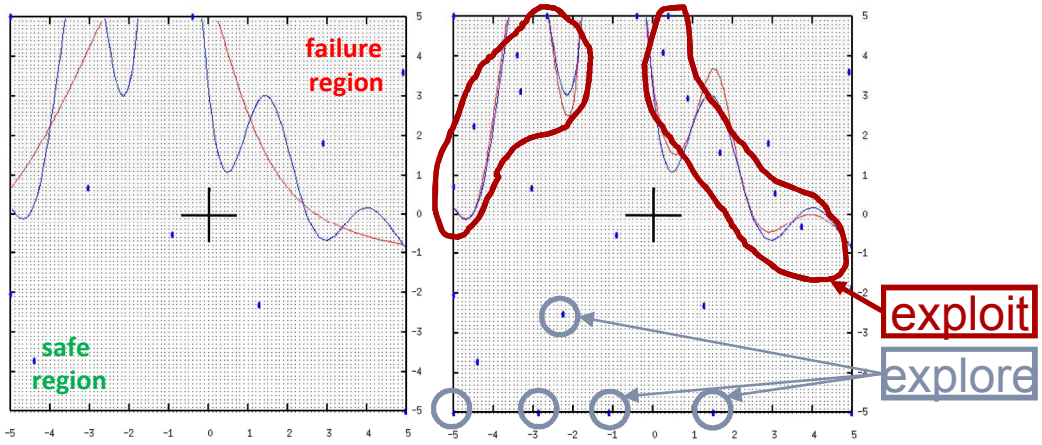
Mean value: make a linearity (and possibly normality) assumption and project; *great for many parameters with efficient derivatives!*

MPP Search Reliability Methods: directly determine input variables which give rise to failure behaviors by solving an optimization problem for a most probable point (MPP) of failure.

Transform the uncertainty problem to an optimization problem: *great for many parameters with efficient derivatives!*

Reliability Methods: How Do They Work?

- Reliability methods try to directly calculate statistics of interest:
  - Make simplifying approximations and/or
  - Recast the UQ as an iterative procedure, such as iteratively refined sampling or as a nonlinearly constrained optimization problem



Mean-value: uses derivatives; make a linearity (and possibly normality) assumption and project

$$\mu_g = g(\mu_x)$$

$$\sigma_g^2 = \sum_i \sum_j Cov_x(i,j) \frac{dg}{dx_i}(\mu_x) \frac{dg}{dx_j}(\mu_x)$$

MPP: solve an optimization problem to directly determine input values giving rise to most probable point of failure

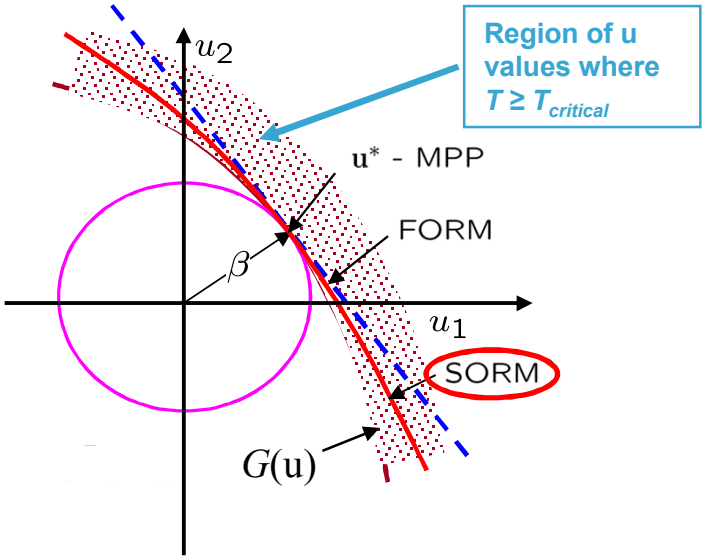
Adaptive: iteratively refine understanding of failure region

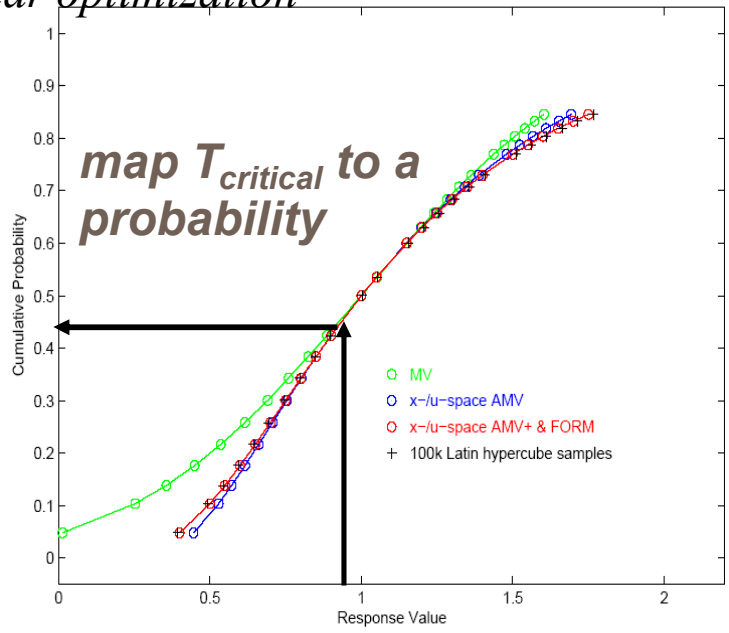
Analytic Reliability: MPP Search

Perform optimization in uncertain variable space to determine Most Probable Point (of response or failure occurring) for *G(u) = T(u)*.

$$\begin{aligned} &\text{minimize} && \mathbf{u}^T\mathbf{u} \\ &\text{subject to} && G(\mathbf{u}) = \bar{z} \end{aligned}$$

All the usual nonlinear optimization tricks apply…

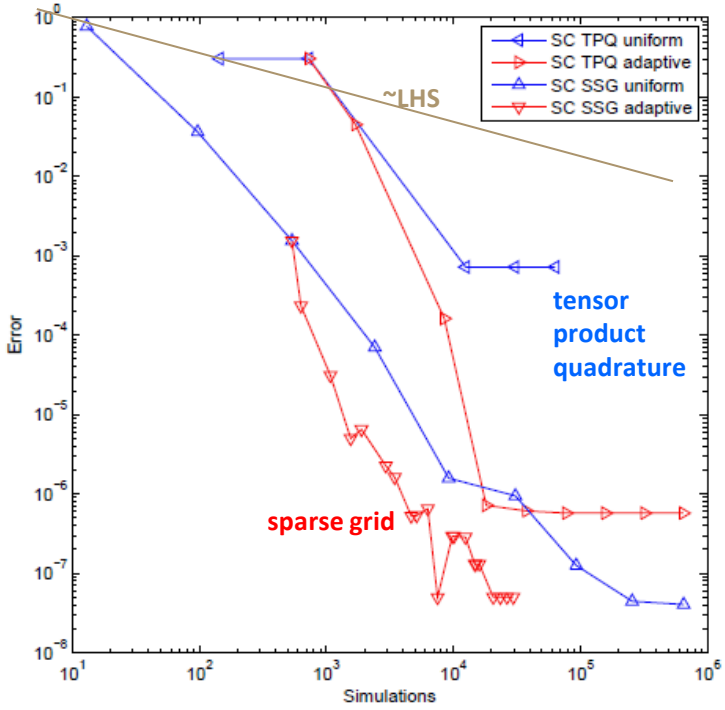




Stochastic Expansions: What Are They?

- General-purpose UQ methods that build UQ-tailored polynomial approximations of the output responses
- Perform particularly well for smooth model responses
- Resulting convergence of statistics can be considerably faster than sampling methods

- Need to specify the Dakota method:
  - Polynomial Chaos (polynomial_chaos): specify the type of orthogonal polynomials and coefficient estimation scheme, e.g., sparse grid or linear regression.
  - Stochastic Collocation (stoch_collocation): specify the type of polynomial basis and the points at which the response will be interpolated; supports piecewise local basis

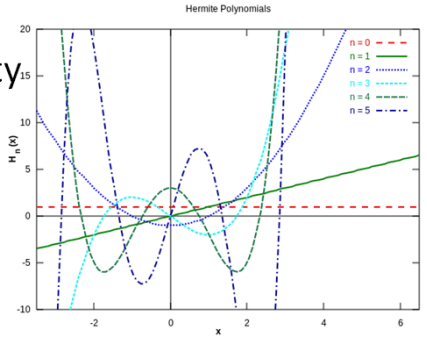


Polynomial Chaos

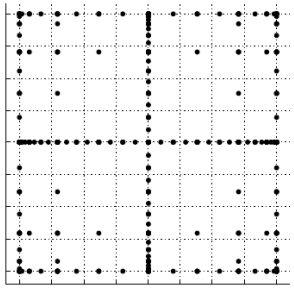
- Uses an orthogonal polynomial basis $\varphi_i(u)$, e.g., Wiener-Askey basis, with Hermite polynomials orthogonal w.r.t. normal density, Legendre polynomials orthogonal w.r.t. uniform density
- Evaluates the model in a strategic way (sampling, quadrature, sparse grids, cubature)…
- …to efficiently approximate the coefficients of an orthogonal polynomial approximation of the response

$$f(u) \approx p(u) = \sum_i c_i \varphi_i(u)$$

- And analytically calculates statistics from the approximation instead of approximating the statistics with MC samples



Hermite Polynomials



Sparse Grid

Application of sensitivities: Regression PCE

- Polynomial Chaos Expansion methods approximate the functional dependence of the simulation response on uncertain model parameters by expansion in an orthogonal polynomial basis.
- There are a variety of ways to generate the coefficients governing a PCE.
- One way is to solve a linear system to compute the coefficients:

$$\hat{u}(x_k) = \bar{u}_k \implies \sum_{i=0}^{P} u_i \Psi_i(x_k) = \bar{u}_k, \quad f(\bar{u}_k, x_k) = 0, \quad k = 0, \ldots, Q$$

  - In Dakota, the points typically are LHS points
    - Quadrature grid – probabilistic collocation
    - Random LHS grid – point collocation
  - System can be ill-conditioned
    - 2x oversampling helps (least squares)
    - Now have a number of specialized regression methods to solve over and underdetermined systems, various compressive sensing algorithms
- **Reduce number of samples by adding derivative equations: reduction is greater for more parameters!**

$$\sum_{i=0}^{P} u_i \Psi_i(x_k) = \bar{u}_k, \quad k = 0, \ldots, Q$$

$$\sum_{i=0}^{P} u_i \frac{\partial \Psi_i}{\partial x}(x_k) = \frac{\partial \bar{u}_k}{\partial x_k}, \quad k = 0, \ldots, Q$$

Application of sensitivities: Regression PCE

Sandia National Laboratories

- Polynomial Chaos Expansion methods approximate the functional dependence of the simulation response on uncertain model parameters by expansion in an orthogonal polynomial basis.
- There are a variety of ways to generate the coefficients governing a PCE.
- One way is to solve a linear system to compute the coefficients:

$$\hat{u}(x_k) = \bar{u}_k \implies \sum_{i=0}^{P} u_i \Psi_i(x_k) = \bar{u}_k, \quad f(\bar{u}_k, x_k) = 0, \quad k = 0, \ldots, Q$$

  - In Dakota, the points typically are LHS points
    - Quadrature grid – probabilistic collocation
    - Random LHS grid – point collocation
  - System can be ill-conditioned
    - 2x oversampling helps (least squares)
    - Now have a number of specialized regression methods to solve over and underdetermined systems, various compressive sensing algorithms
- Reduce number of samples by adding derivative equations

$$\sum_{i=0}^{P} u_i \Psi_i(x_k) = \bar{u}_k, \quad k = 0, \ldots, Q$$

$$\sum_{i=0}^{P} u_i \frac{\partial \Psi_i}{\partial x}(x_k) = \frac{\partial \bar{u}_k}{\partial x_k}, \quad k = 0, \ldots, Q$$

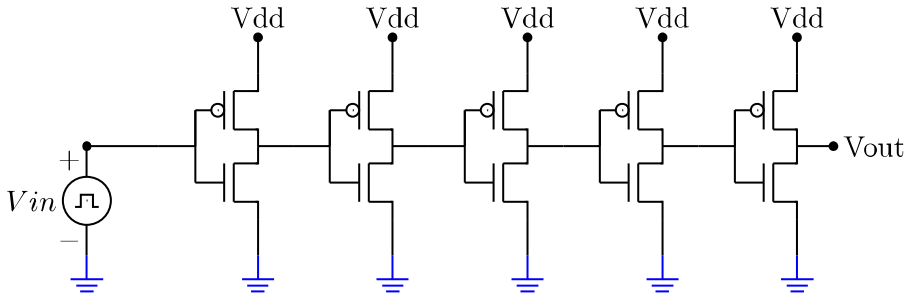
Gradient-Enhanced PCE Results

Eric Keiter, Sandia National Laboratories
11th International Conference on Scientific Computing in Electrical Engineering

Transient Sensitivities Example
5-stage CMOS inverter

Sandia National Laboratories

- BSIM6 Model
- 5 stages.
- Each stage adds to signal delay

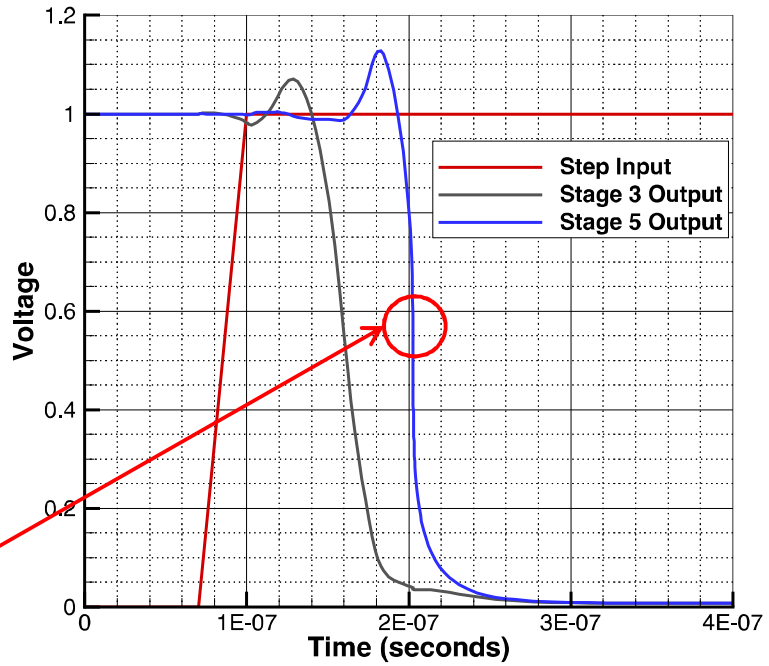


- Capacitive effects are the main delay contributors
  - PMOS and NMOS oxide thickness are the design parameters.
- Step input (Vin),Output at Vout.
- Objective function: Generalized Elmore delay.

$$O = \texttt{Elmore Delay} = \frac{\int_0^T g'_A(t) \cdot t \cdot dt}{\int_0^T g'_A(t) dt}$$

$$g_A(t) = \texttt{Vout}$$

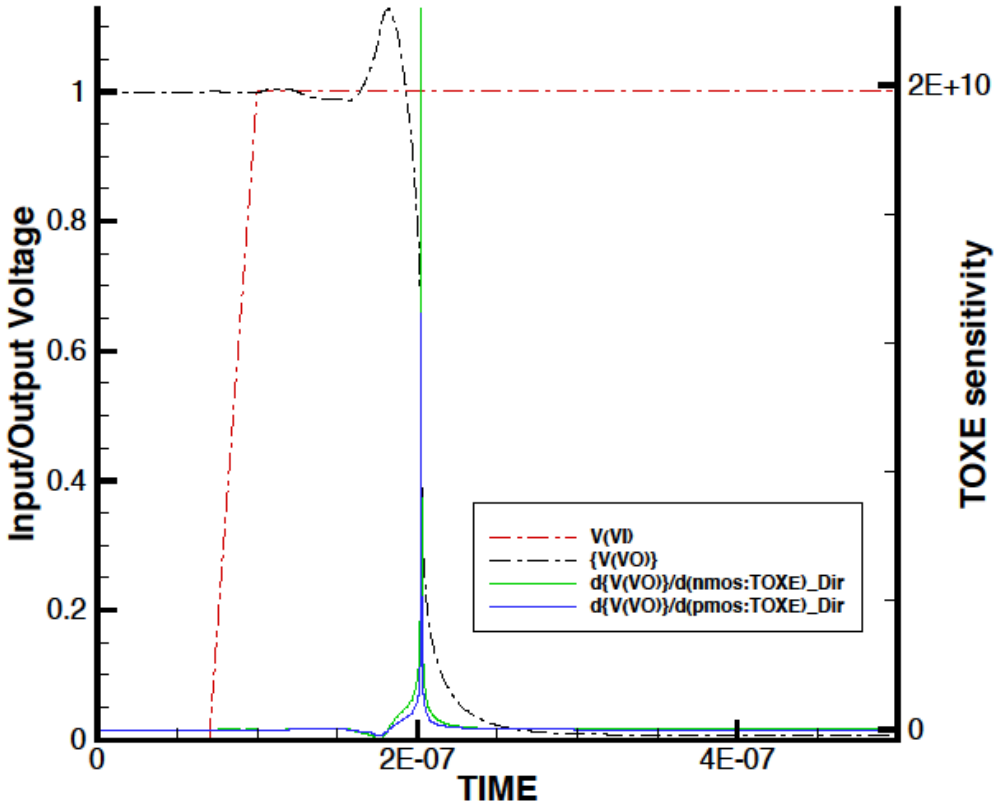
- Elmore delay should give the approximate time for signal rise.



reduced objective (Elmore) = 2.078232472421e-007
reduced objective sensitivity for param dV(VO)d(nmos:TOXE)_Dir = 3.580676543124e+001
reduced objective sensitivity for param dV(VO)d(pmos:TOXE)_Dir = 1.962028129681e+001

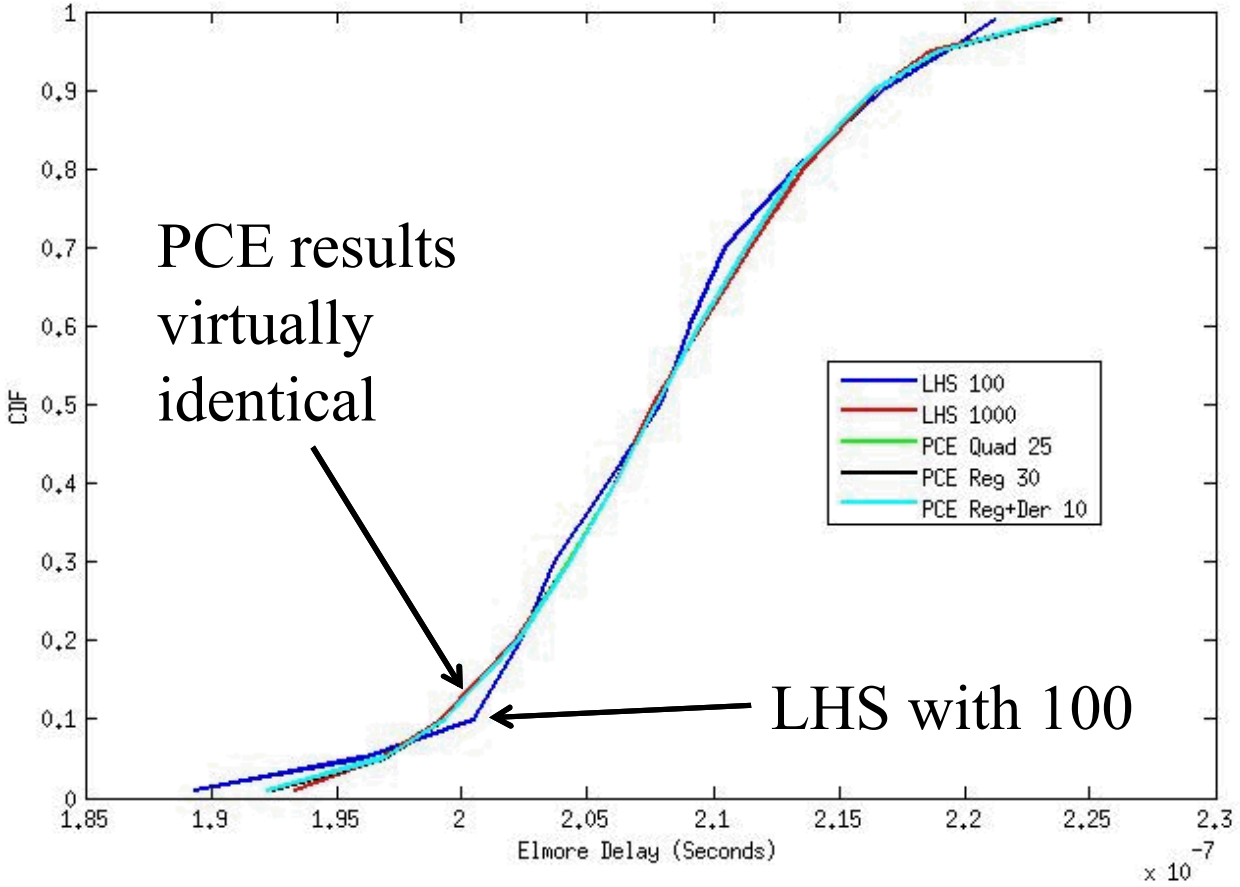
Regression PCE Results

- Simple CMOS circuit which is an inverter chain that uses the BSIM6 transistor device.
- Five invertors
- Want to mimic applications where signal delay is the important metric
- NMOS and PMOS oxide thicknesses are the uncertain parameters: we assume they are normal with std. deviation = 10 percent of nominal values



Regression PCE Results

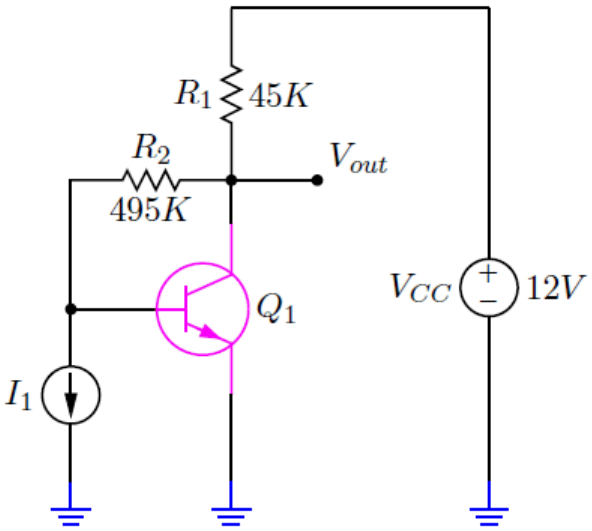
Inverter Circuit



	Mean	Std. Dev.
100 LHS	2.0781E-07	6.6309E-09
1000 LHS	2.0782E-07	6.6935E-09
25 PCE Quad	2.0783E-07	6.6954E-09
30 PCE Regression	2.0783E-07	6.7131E-09
10 PCE Reg+Der	2.0782E-07	6.7035E-09

PCE results accurate with 10 function evaluations and derivatives.
All PCE results very similar to 1000 LHS results.

Common Emitter Amplifier

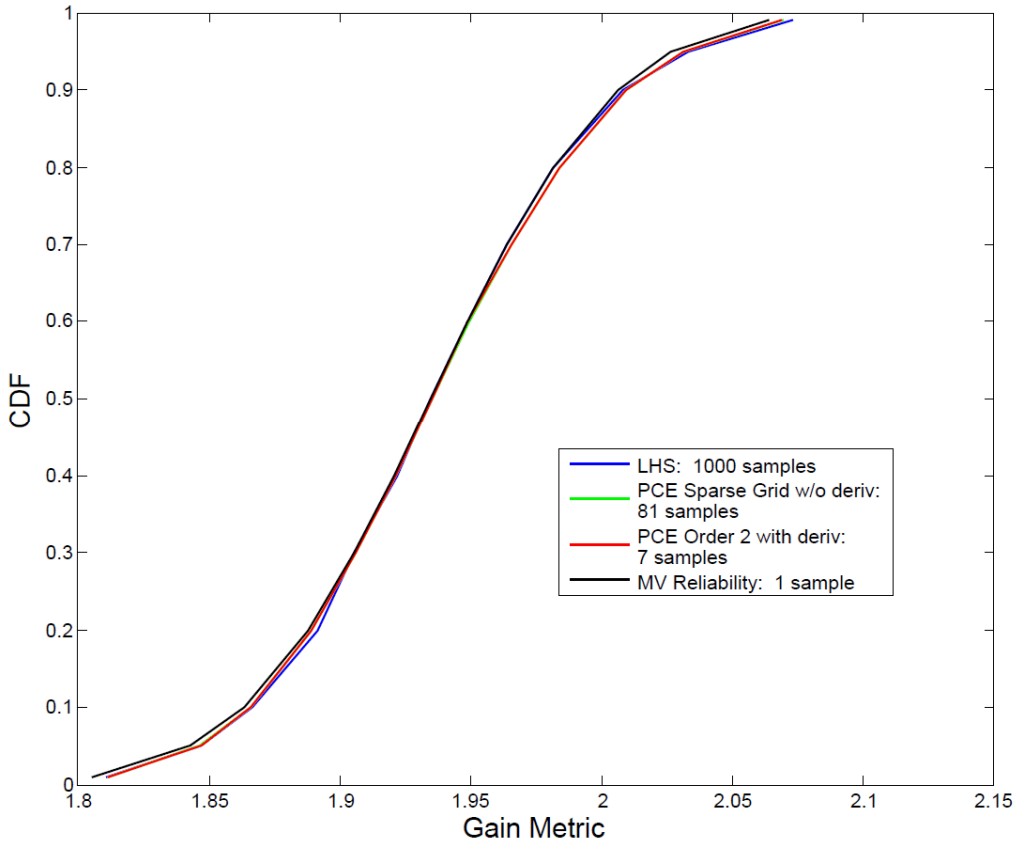


Common Emitter Amplifier. Taken from the Spice OPUS website: http://fides.fe.uni-lj.si/spice/opt001.html

- The purpose of this circuit is to provide voltage and current gain to an input signal.
- The output metric of interest is the voltage gain of this circuit when the input current source reaches its maximum value.
- Five input parameters were varied, treated as normal distributions:
  - Transitor parameters IS, BF
  - Resistances of R1 and R2
  - Temperature

Input Variable	Mean	Std. Deviation
IS	8.35E-14	4.175E-15
BF	150	7.5
R1	200	10
R2	150	7.5
Temp	27	1.35

Common Emitter Amplifier



- The Cumulative Distribution Function (CDF) shows the probability that the Gain Metric is less than a particular value on the x-axis, X:
- CDF(X) = Prob(GainMetric<X)
- CDF varies between 0 and 1
- In this problem, the CDF traces are nearly the same, especially near the 50th percentile
- Slight variation in CDF results in the tails, which is to be expected

Common Emitter Amplifier

Sandia National Laboratories

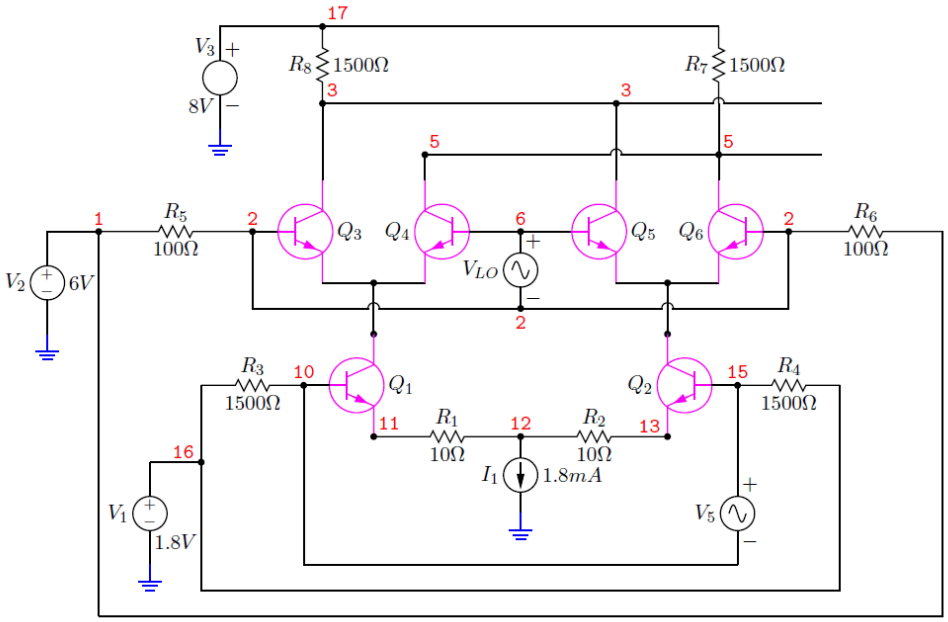
- Table of mean and 95th percentile estimates shows that ***comparable results to 1000 samples can be achieved with very few points with their gradients:*** 7 for PCE or 1 for Mean Value reliability!

Number of Samples and UQ Method	Mean Estimate	95th Percentile Estimate
1000 LHS	1.9365	2.0329
81 PCE – Quadrature Points	1.9366	2.0313
7 PCE – Regression with derivatives	1.9366	2.0313
1 Mean Value Reliability	1.9347	2.0263

- Importance of uncertain inputs in contributing to output (gain metric) variance

Variable	Main Effects Index
IS	0.104
BF	0.003
R1	0.255
R2	0.637
TEMP	0.0009

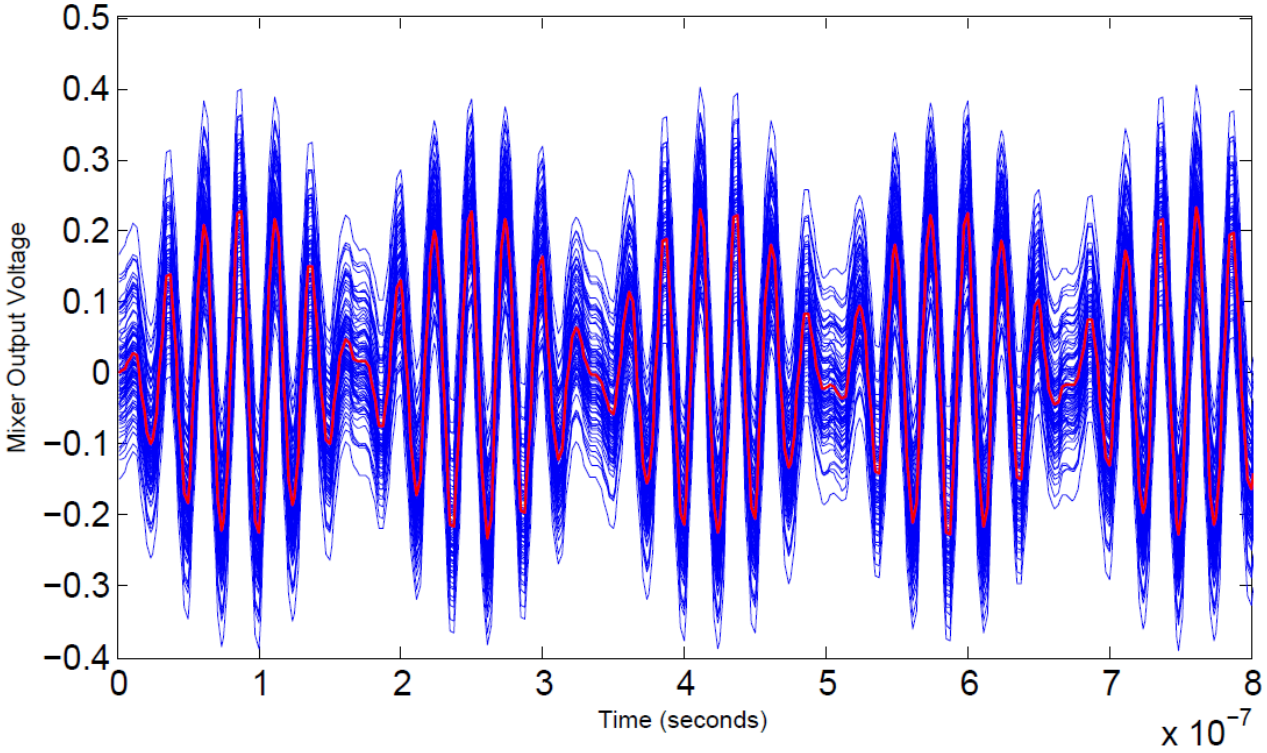
Gilbert Cell Mixer



- Output is the difference between the collector voltages of Q4/Q6 and Q3/Q5
- Eight inputs were varied for this study:
  - R1, R2 ~N(10,1)
  - R5,R6 ~N(100,5)
  - R3,R4,R7,R8 ~ N(1500,50)

- Input resistance uncertainties are not realistic: large to demonstrate uncertainty propagation process

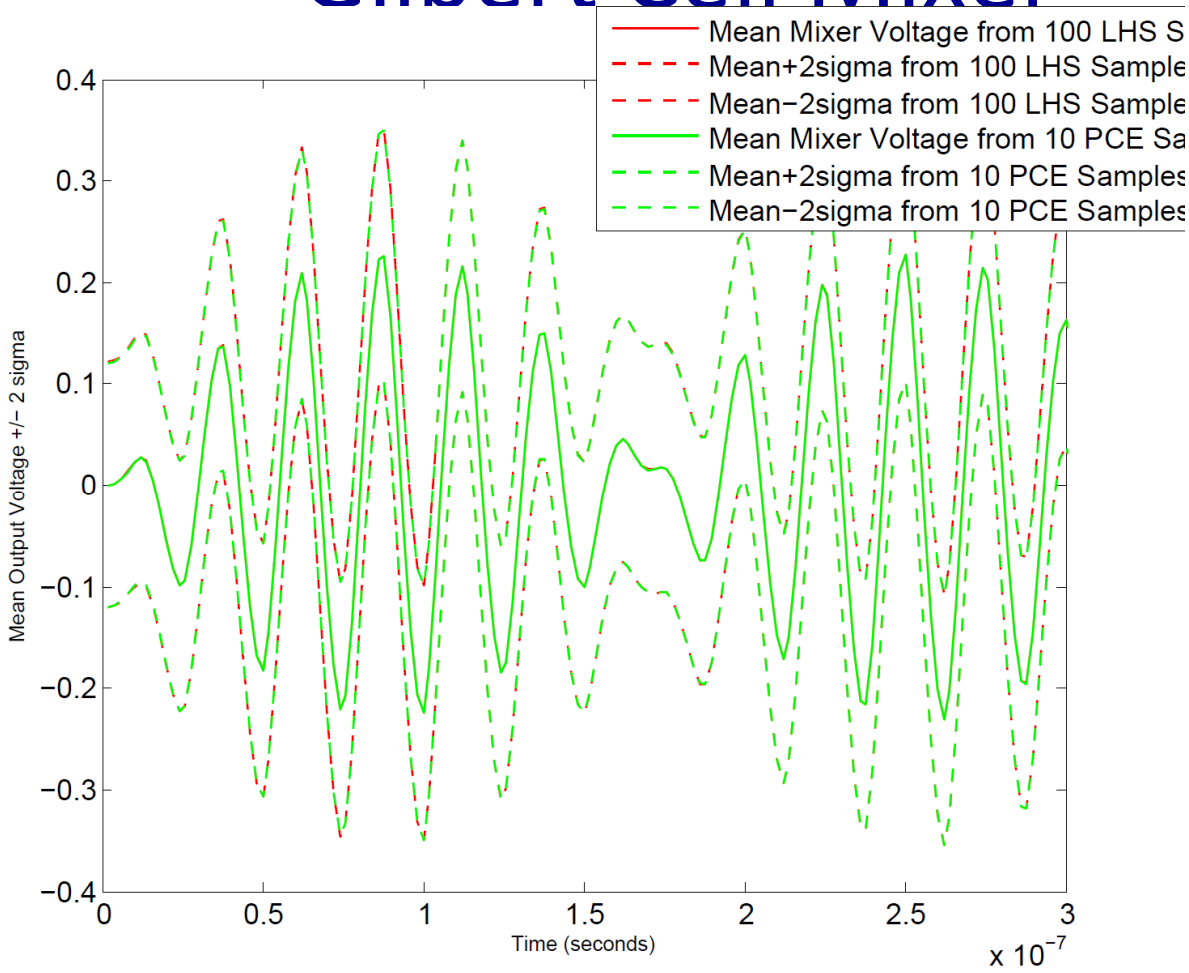
Gilbert Cell Mixer



Ensemble of 100 LHS Samples (blue) show the variation in the results. The mean of the sample results is shown in red. Note the data was taken at time steps of 0.2E-8 seconds and only a few cycles are shown to emphasize the variability.

The overall mean of the mixer voltage is near zero: 7.0E-4 volts.

Gilbert Cell Mixer



Moments of the output voltage (mean +/- 2σ) from PCE with 10 samples and LHS with 100 samples are virtually identical as a function of time

Concluding Remarks

- Gradient-enhanced PCE shows some promise
- At least for smooth problems.
- Implementing adjoint sensitivities is challenging, especially in transient.

- All methods described here are available open-source in Xyce and Dakota.

Extra slides

ADMS-Xyce

ADMS = *Automatic Device Model Synthesizer*

Verilog-A: industry standard format for new models:e.g. VBIC, Mextram, EKV, HiCUM, etc

ADMS translates Verilog-A to Xyce-compliant C/C++ code;

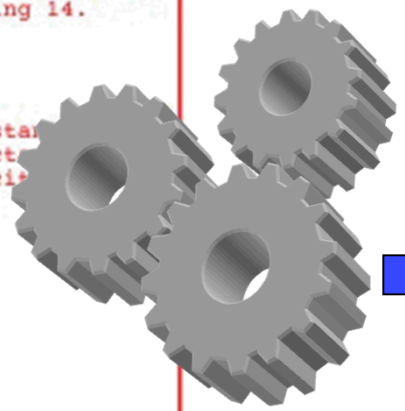
API automatically handles data structures, matrices, tedious details.

ities via Sacado automatic differentiation

ackend to include Stocahstic Expansions via Stokhos.

```
1 // Series RLC
2 // Version 1a, 1 June 04
3 // Ken Kundert
4 //
5 // Downloaded from The Designer's Guide Community
6 // (www.designers-guide.org).
7 // Taken from "The Designer's Guide to Verilog-AMS"
8 // by Kundert & Zinke. Chapter 3, Listing 14.
9
10 `include "disciplines.vams"
11
12 module series_rlc2 (p, n);
13     parameter real r=1000;      // resista
14     parameter real l=1e-9;      // induct
15     parameter real c=1e-6;      // capaci
16     inout p, n;
17     electrical p, n, i;
18     branch (p, i) rl, (i, n) cap;
19
20     analog begin
21     V(rl) <+ r*I(rl);
22     V(rl) <+ ddt(l*I(rl));
23     I(cap) <+ ddt(c*V(cap));
24     end
25 endmodule
```

Verilog-A



Run admsXyce

```
// -- code converted from analog/code block// I(
((V(p,internal1)/R))staticContributions[admsNodeI
((probeVars[admsProbeID_V_p_internal1])/instanceP
deID_internal1] -=
((probeVars[admsProbeID_V_p_internal1])/instanceP
((probeVars[admsProbeID_V_internal1_internal2])*i
nternal1,internal2) <+
(CapacitorCharge))dynamicContributions[admsNo
(CapacitorCharge);dynamicContributions[admsNodeID
(CapacitorCharge);InductorCurrent = (probeVars[ad
V(internal2,n) <+
((L*ddt(InductorCurrent)))dynamicContributions[ad
(instancePar_L*(InductorCurrent));
```

C++ code snippet

(actual Xyce file is 1500 lines)

ADMS-Xyce Tidbits

- ADMS = *Automatic Device Model Synthesizer*
  - Open source tool, developed by Motorola/Freescale/NXP.
- ADMS-Xyce back-end development started in 2007
  - Development focused on custom back-end for Xyce
  - Major development effort for Xyce project.
- ADMS-Xyce now supports dynamic linking of models (*.so files) in Xyce
  - Allows Xyce users to write models in Verilog-A in real time, and get all benefits.

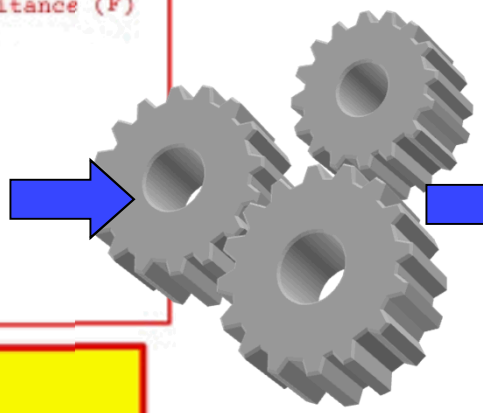
- ADMS-converted models in Xyce:
  - MOSFET models: EKV PSP (version 103); JunCap (diode model), but part of PSP; BSIM-CMG 107 (FinFET); BSIM6; MVS (MIT virtual source model)
  - Bipolar models: VBIC; FBH HBT; MEXTRAM

Main Punchline: *All industrial strength, industry models are published in Verilog-A. Therefore, all of these models MUST come into Xyce via ADMS or some other model compiler like ModSpec.*

ADMS-Xyce
Development Status and Plans

```
1  // Series RLC
2  // Version 1a, 1 June 04
3  // Ken Kundert
4  //
5  // Downloaded from The Designer's Guide Community
6  // (www.designers-guide.org).
7  // Taken from "The Designer's Guide to Verilog-AMS"
8  // by Kundert & Zinke. Chapter 3, Listing 14.
9
10 `include "disciplines.vams"
11
12 module series_rlc2 (p, n);
13     parameter real r=1000;      // resistance (Ohms)
14     parameter real l=1e-9;      // inductance (H)
15     parameter real c=1e-6;      // capacitance (F)
16     inout p, n;
17     electrical p, n, i;
18     branch (p, i) rl, (i, n) cap;
19
20     analog begin
21     V(rl) <+ r*I(rl);
22     V(rl) <+ ddt(l*I(rl));
23     I(cap) <+ ddt(c*V(cap));
24     end
25 endmodule
```

Verilog-A



NextGen (ATDM)
Kokkos
NVIDIA

Embedded UQ
Stokhos
$$f(\theta) = \sum_{k=0}^{\infty} F_{pk}\Phi_k(\epsilon)$$

Planned

Dynamic Linking
`*.so` files

Param Sensitivities
Sacado
$$\frac{d\vec{f}(\vec{x})}{dp}, \frac{d\vec{q}(\vec{x})}{dp}$$

DAE support
Sacado (Jacobian)
$$\frac{d}{dt}\vec{q}(\vec{x}(t)) + \vec{f}(\vec{x}(t), \vec{u}(t)) = \vec{0}$$

Implemented

Verilog-A (high level modeling language) Input

Run admsXyce

Ready-to-compile C++ code

Sandia National Laboratories

Polynomial Chaos Expansions (PCE)

- Utilizes series of orthogonal polynomials to facilitate stochastic analysis
- Considered here are the Hermite polynomials:

$$\Phi_0(\xi) = 1,\ \Phi_1(\xi) = \xi,\ \text{and}\ \Phi_2(\xi) = \xi^2 - 1, \ldots$$

- If $f(t,\theta)$ is a random signal, then expand: $f(t,\theta) = \sum_{k=0}^{\infty} F_{pk}\Phi_k(\xi)$

$$F_{pk} = (1/\left\langle \Phi_k^2(\xi) \right\rangle) \int_{-\infty}^{\infty} f(t,\theta)\Phi_k(\xi)W(\xi)\mathrm{d}\xi$$

- Notation: θ : random event

  $\xi = \xi(\theta)$: random variable for Gaussian distribution

  $W(\xi)$: weighting function, here probability density function of ξ

- For other probability density functions (pdfs), other polynomials available

Analogy: Fourier Series and Polynomial Chaos

method	Harmonic Balance (HB) in Xyce: Fourier series	Proposed Polynomial Chaos (PCE) in Xyce: Hermite polynomial chaos expansion
object	periodic signal $f(t,T)$ the period is T and $\omega = 2\pi/T$	random signal $f(t,\theta)$ the random input is $\xi(\theta) \sim N(0,1)$
orthogonal basis	complex exponential fcts. $\{e^{jk\omega t}\}$ $\langle e^{jk\omega t}, e^{jn\omega t}\rangle =$ $\int_{-\infty}^{\infty} e^{jk\omega t} e^{jn\omega t} W(t)dt = T\delta_{kn}$	Hermite polynomial fcts. $\{\Phi_k(\xi(\theta))\}$ $\langle \Phi_k(\xi), \Phi_n(\xi)\rangle =$ $\int_{-\infty}^{\infty} \Phi_k(\xi)\Phi_n(\xi)W(\xi)d\xi = \langle \Phi_k^2(\xi)\rangle \delta_{kn}$
weighting function	$W(t) = u(t_0) - u(t_0 - T)$	$W(\xi) = e^{-\xi^2/2}/\sqrt{2\pi}$
expansion	$f(t) = \sum_{k=-\infty}^{\infty} F_{fk} e^{jk\omega t}$	$f(t,\theta) = \sum_{k=0}^{\infty} F_{pk}\Phi_k(\xi)$
expansion coefficient, random mode	Frequency component: $F_{fk} = (1/T)\int_{-\infty}^{\infty} f(t)e^{-jk\omega t}W(t)dt$	component at random mode k : $F_{pk} = (1/\langle \Phi_k^2(\xi)\rangle)\int_{-\infty}^{\infty} f(t,\theta)\Phi_k(\xi)W(\xi)d\xi$
mean	F_{f0}	F_{p0}

HB device transformation:

Original Device Model → FFT → Freq Domain Device Model

PCE device transformation:

Original Device Model → ADMS/Xyce + Stokhos → Stochastic Device Model

Both HB and PCE will result in large block linear systems

Table from: K. Strunz and Q. Su. "Stochastic formulation of SPICE-type electronic circuit simulation with polynomial chaos." ACM Trans.Model. Comput. Simul. 18, 4, Article 15, Sept. 2008.