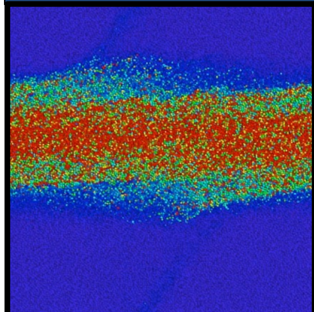
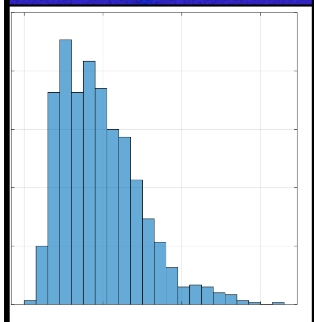
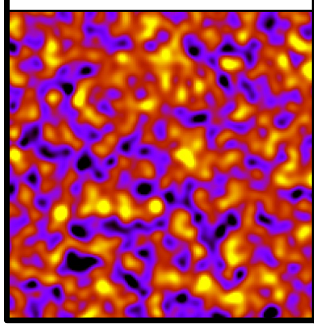
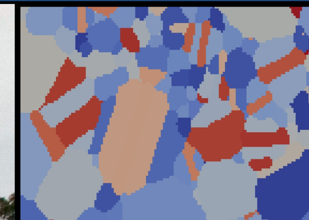
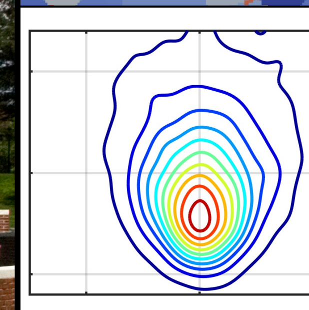
USACM Conference on Uncertainty Quantification in Computational Solid and Structural Materials Modeling

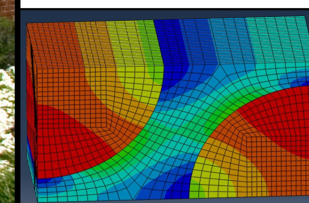


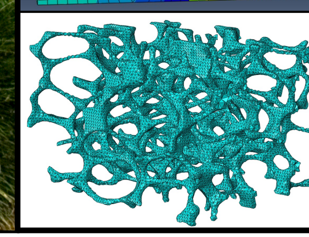












Confirmed Speakers

Wilkins Aquino, Duke
Zdenek Bazant, Northwestern
George Deodatis, Columbia
Phillipe Geubelle, Illinois
Roger Ghanem, Southern California
George Karniadakis, Brown
Jarek Knap, ARL

Marisol Koslowski, Purdue
Jia-Lian Le, Minnesota
Jie Lie, Tongji
Yongming Liu, Arizona State
David McDowell, Georgia Tech
Michael Ortiz, Cal Tech

Martin Ostoja-Starzewski, Illinois
Simon Phillpot, Florida
Pedro Ponte Casteneda
Jim Stewart, Sandia
Ben Thacker, Southwest Research Institute
Clayton Webster, ORNL

Software - UQpy

A collection of Python modules used for uncertainty quantification and propagation

- Includes commonly applied methods and new developments

- Developed collaboratively by members of SURG
  - Authors: Michael D. Shields, Dimitris Giovanis
  - Contributors: Jiaxin Zhang, Lohit Vandanapu, Mohit Chauhan, Aakash Bangalore Satish

- Version control through git (requires Python 3)
  - Alpha Version 0.1.0 available via GitHub (https://github.com/SURGroup/UQpy)
  - Version 1.0.0 expected for public release in May 2018

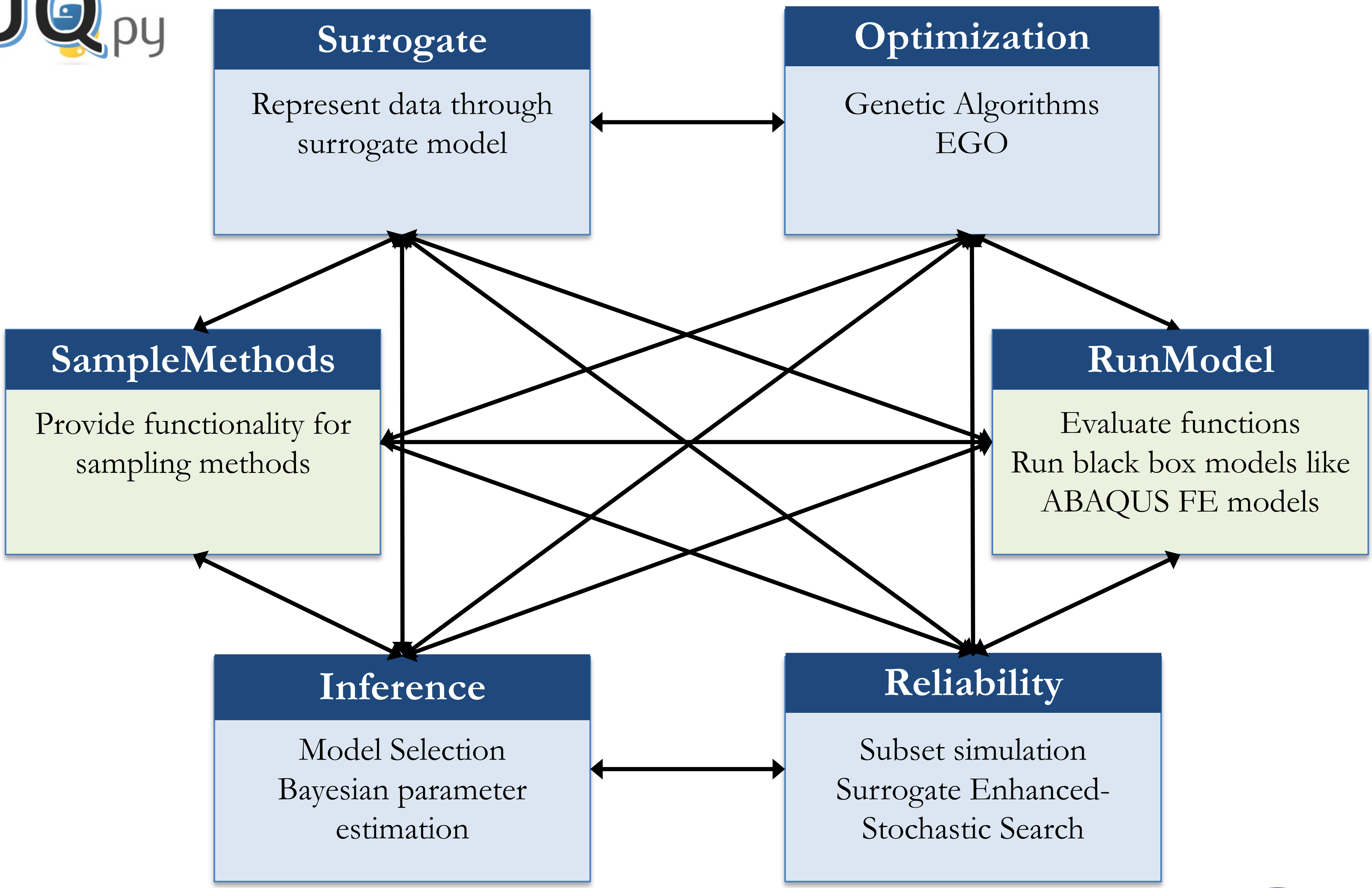
```
$git clone https://github.com/SURGroup/UQpy.git
$cd UQpy/
$pip install -r requirements.txt
$python setup.py install
```

```
from UQpy.SampleMethods import MCMC

def Rosenbrock(x):
    return np.exp(-(100*(x[1]-x[0]**2)**2+(1-x[0])**2)/20)

x = MCMC(dimension=2, pdf_proposal_type='Normal', pdf_proposal_width=1, pdf_target_type='joint_pdf',
        pdf_target=Rosenbrock, algorithm='MMH', jump=100, nsamples=100, seed=None)
```

Software – UQpy Modules



SURG Shields Uncertainty Research Group

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

Software – UQpy Functionality

Current Functionality:

- Monte Carlo Simulation
- Latin hypercube sampling
- Markov Chain Monte Carlo
  - Metropolis-Hastings
  - Modified Metropolis-Hastings
  - Affine Invariant Ensemble
- Stratified Sampling
- Partially Stratified Sampling
- Stochastic Reduced Order Models
- Subset Simulation
- Bayesian Inference
  - Multimodel inference
  - Bayesian parameter estimation

Functionality Under Development:

- Stratified Sampling
  - Refined Stratified Sampling
  - Gradient-Enhanced Refined Stratified Sampling
  - Latinized Stratified Sampling
  - Latinized Refined Stratified Sampling
  - Targeted Random Sampling
- Stochastic Process Simulation
  - Spectral Representation Method
  - Karhunen-Loeve Expansion
  - Iterative Translation Approximation Method
  - Bispectral Representation Method
- Importance Sampling
- FORM/SORM
- Stochastic Collocation
  - SSC
  - VSSC
  - Grassmann Refinement
- Surrogate Modeling
  - Kriging
    - AK-MCS
    - Multimodel Kriging
  - ANNs
  - Polynomial Chaos
- Optimization
  - EGO
  - Genetic Algorithms
- Surrogate Enhanced Stochastic Search
- Sensitivity Analysis

MS81: Theory and Simulation of Failure Probabilities and Rare Events

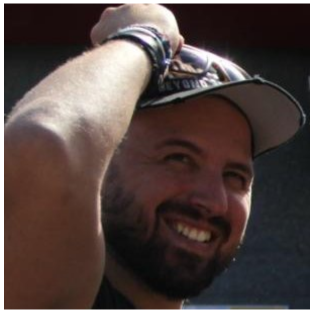
Ensemble MCMC Samplers for Failure Probability Estimation with Subset Simulation

Michael D. Shields
Assistant Professor
Dept. of Civil Engineering
Dept. of Materials Science and Engineering
Johns Hopkins University

Acknowledgements

Dr. V.S. Sundar

Postdoctoral Researcher,
Dept. of Radiology, Center for Multimodal Imaging and Genetics,
University of California San Diego



Dr. Dimitris Giovanis

Postdoctoral Researcher,
Dept. of Civil Engineering
Johns Hopkins University

Jiaxin Zhang

Ph.D. Candidate,
Dept. of Civil Engineering
Johns Hopkins University

Background

Subset Simulation

$$\Psi = \left[G(\boldsymbol{U}) \le 0 \right]$$

$$\Psi_1 \supset \Psi_2 \supset \cdots \supset \Psi_M = \Psi$$

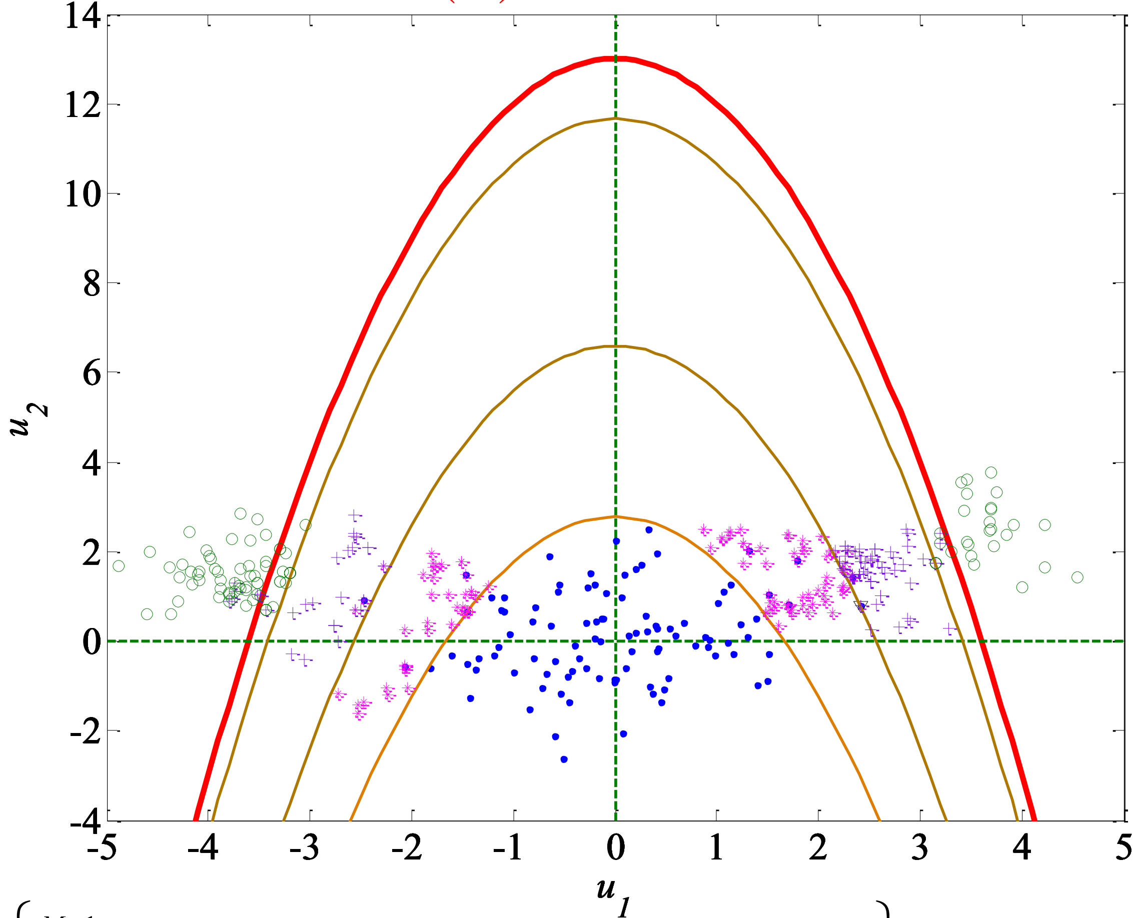
$$\Psi_k = \bigcap_{i=1}^{k} \Psi_i, k = 1, 2, \cdots, M$$

$$P_F = P(\Psi_M) = P(\Psi_1) \prod_{i=1}^{M-1} P(\Psi_{i+1} \mid \Psi_i)$$

- MCMC is used to evaluate the conditional $P(\Psi_{i+1}|\Psi_i)$

- Modified MH algorithm

$$G(\boldsymbol{U}) = 13 - U_1^2 - U_2$$



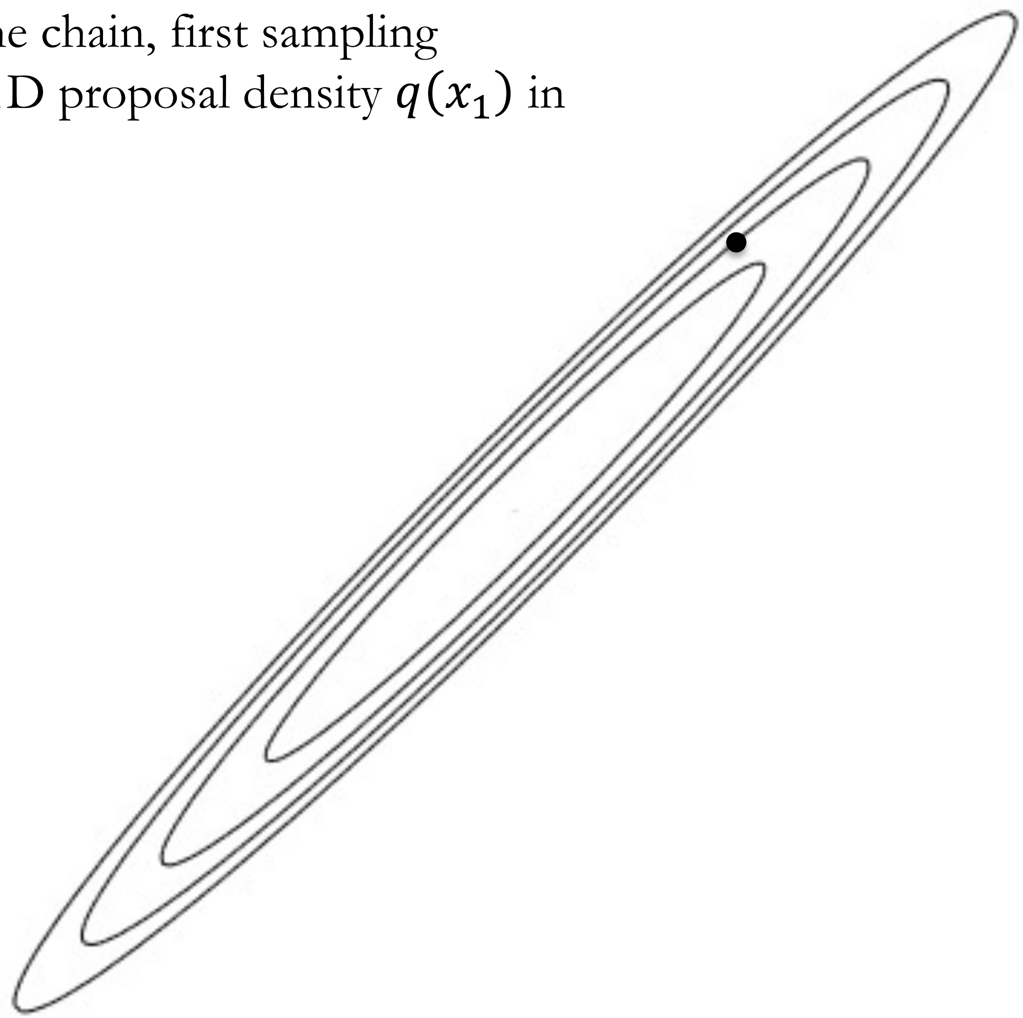
$$\hat{P}_F = \left\{ \frac{1}{n_1} \sum_{j=1}^{n_1} I\left[G_1(\boldsymbol{U}_j) \le 0 \right] \right\} \left\{ \prod_{i=1}^{M-1} \frac{1}{n_{i+1}} \sum_{j=1}^{n_{i+1}} I\left[G_{i+1}(\boldsymbol{U}_j) \le 0 \middle| G_i(\boldsymbol{U}_j) \le 0 \right] \right\}$$

Au, S. and Beck, J. (2001). *Probabilistic Engineering Mechanics*. 16: 263-277

Modified Metropolis-Hastings

Consider the current state of the MC

- To propagate the chain, first sampling according to a 1D proposal density $q(x_1)$ in dimension x_1

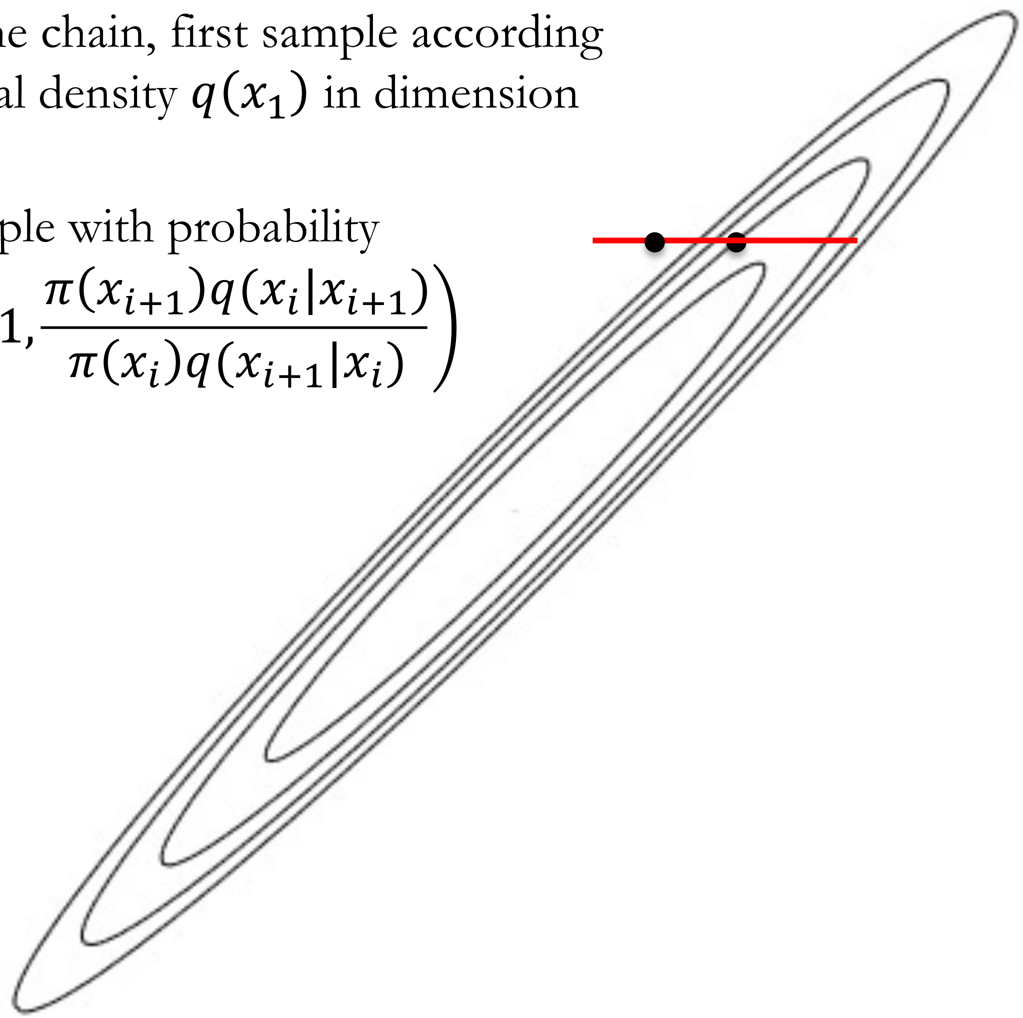


Modified Metropolis-Hastings

Consider the current state of the MC

- To propagate the chain, first sample according to a 1D proposal density $q(x_1)$ in dimension x_1
- Accept the sample with probability

$$\alpha = \min\left(1, \frac{\pi(x_{i+1})q(x_i|x_{i+1})}{\pi(x_i)q(x_{i+1}|x_i)}\right)$$

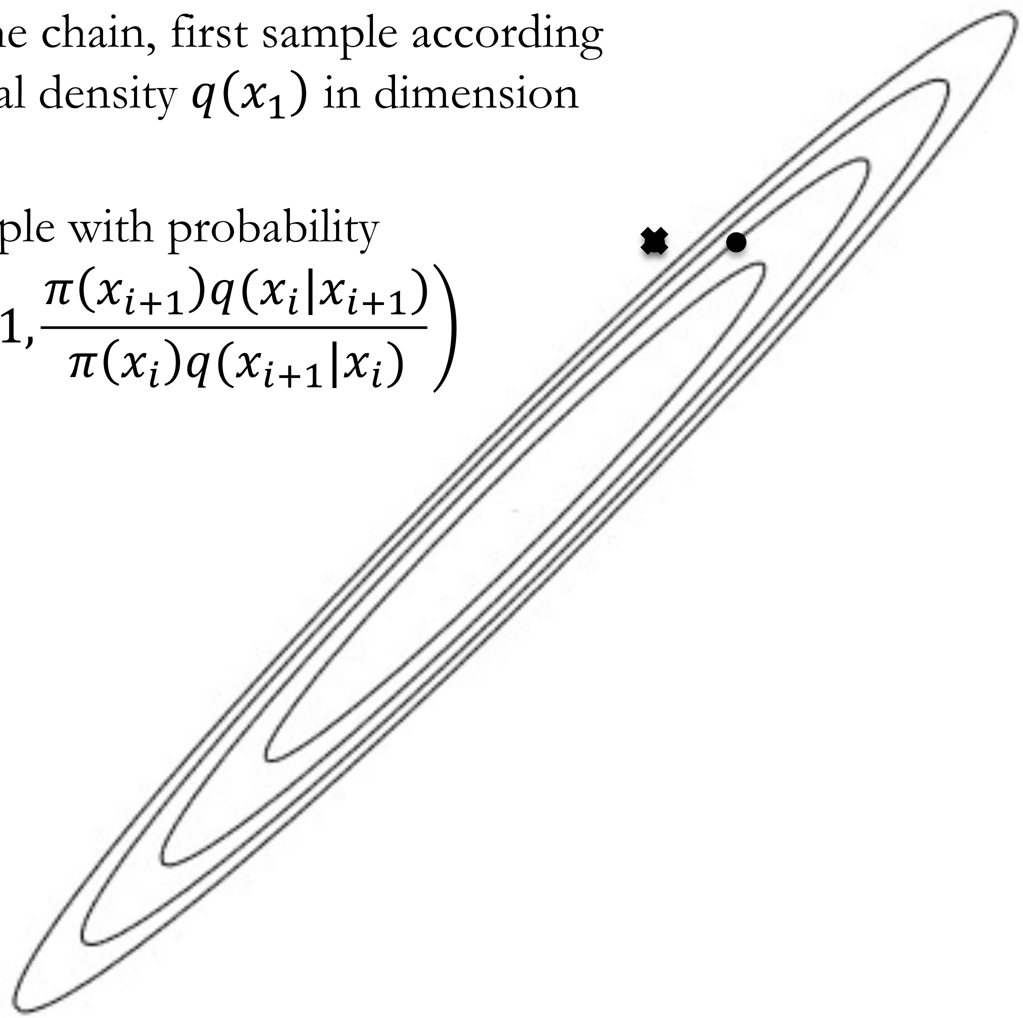


Modified Metropolis-Hastings

Consider the current state of the MC

- To propagate the chain, first sample according to a 1D proposal density $q(x_1)$ in dimension x_1
- Accept the sample with probability

$$\alpha = \min\left(1, \frac{\pi(x_{i+1})q(x_i|x_{i+1})}{\pi(x_i)q(x_{i+1}|x_i)}\right)$$

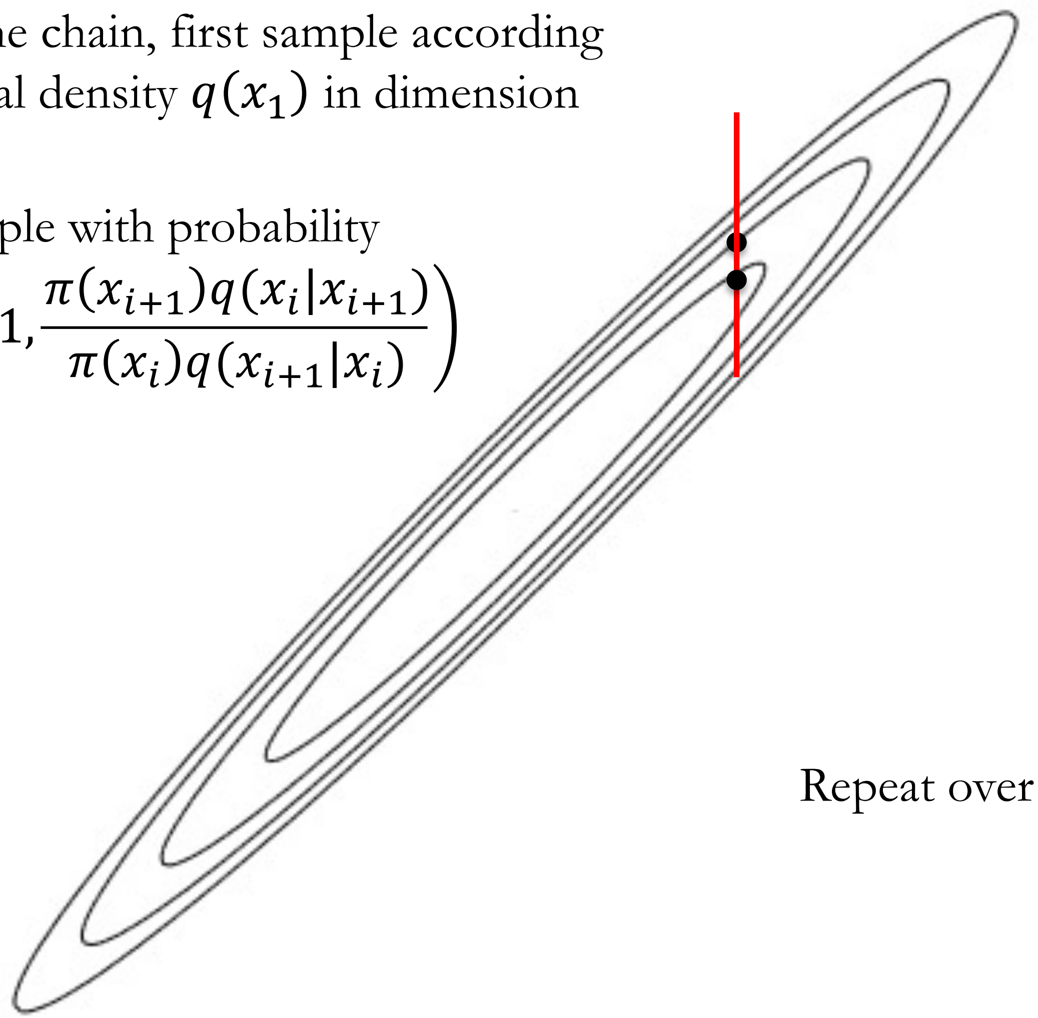


Modified Metropolis-Hastings

Consider the current state of the MC

- To propagate the chain, first sample according to a 1D proposal density $q(x_1)$ in dimension x_1
- Accept the sample with probability

$$\alpha = \min\left(1, \frac{\pi(x_{i+1})q(x_i|x_{i+1})}{\pi(x_i)q(x_{i+1}|x_i)}\right)$$



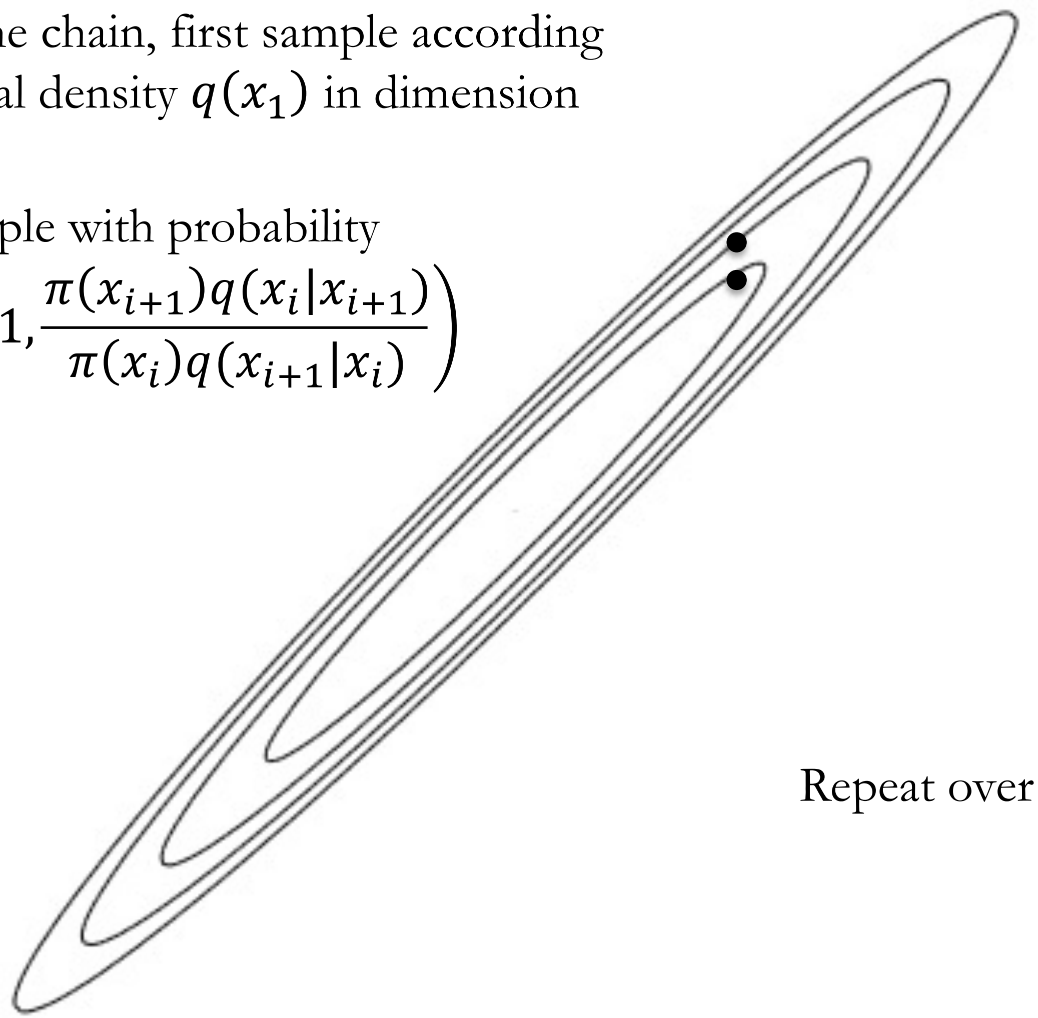
Repeat over all dimensions

Modified Metropolis-Hastings

Consider the current state of the MC

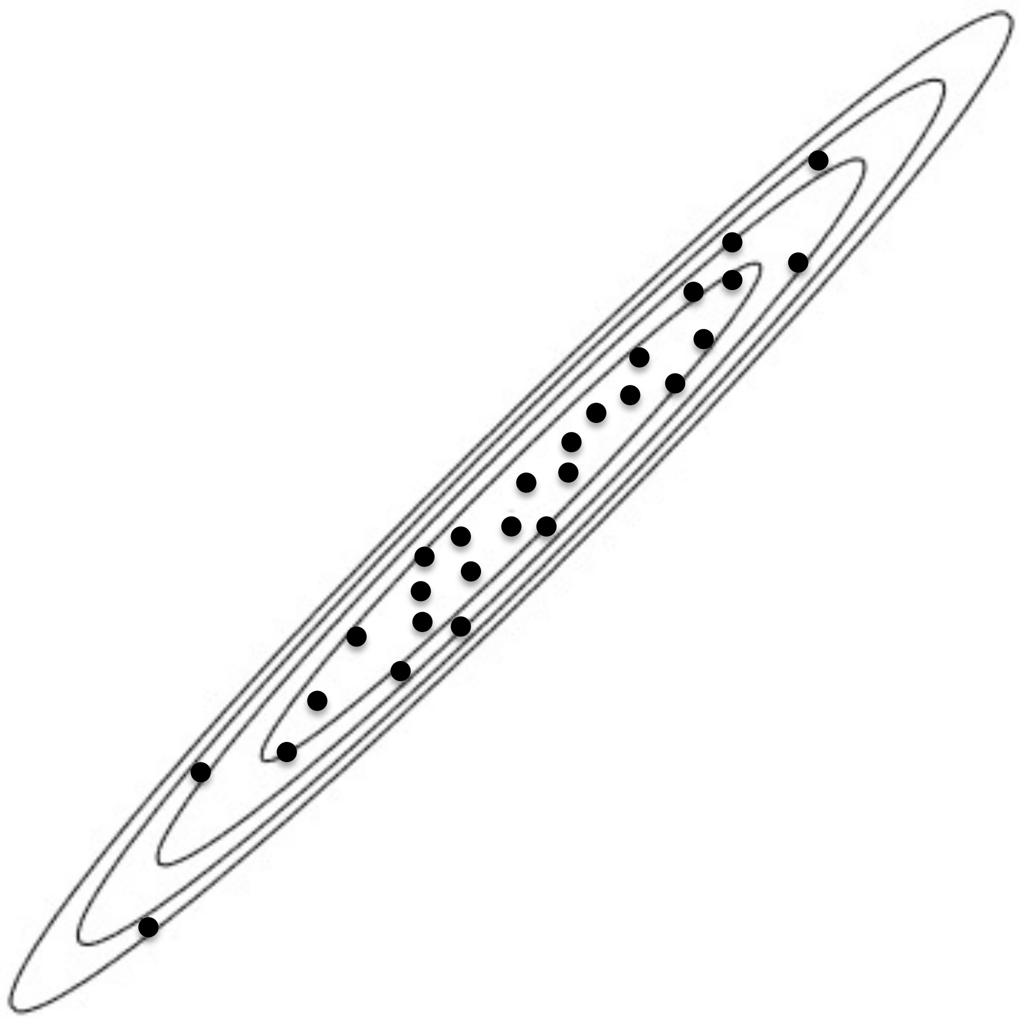
- To propagate the chain, first sample according to a 1D proposal density $q(x_1)$ in dimension x_1
- Accept the sample with probability

$$\alpha = \min\left(1, \frac{\pi(x_{i+1})q(x_i|x_{i+1})}{\pi(x_i)q(x_{i+1}|x_i)}\right)$$



Repeat over all dimensions

Affine Invariant Ensemble Sampler – Stretch Moves

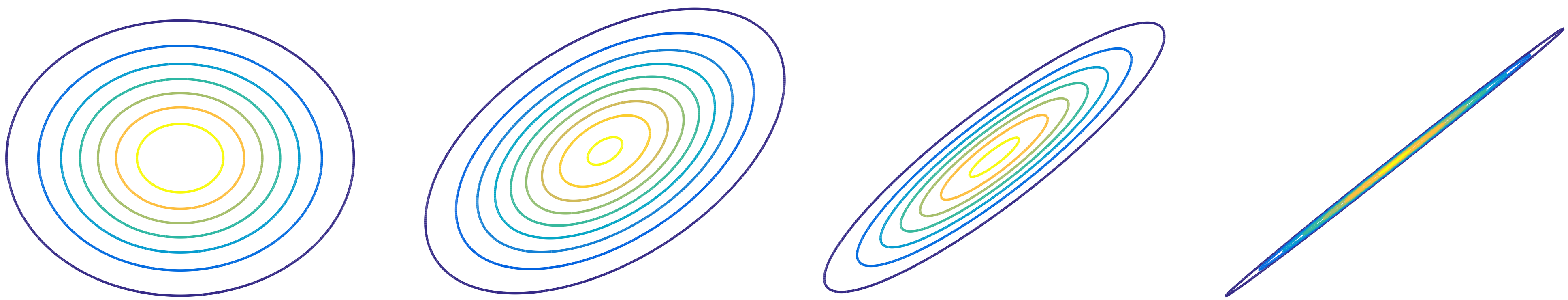


Degeneracy

Convention holds that subset simulation is typically performed in a standard normal space

- For many applications, this works very well
- Component-wise modified Metropolis-Hastings (MMH) algorithms are incredibly robust in this case

$\rho = 0$ | $\rho = 0.5$ | $\rho = 0.9$ | $\rho = 0.999$



MMH Good | MMH Good | MMH Good | MMH Not Good

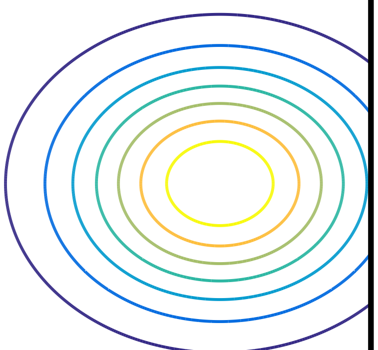
As the distribution degenerates, the correlation length of the MMH samples increases

Degeneracy

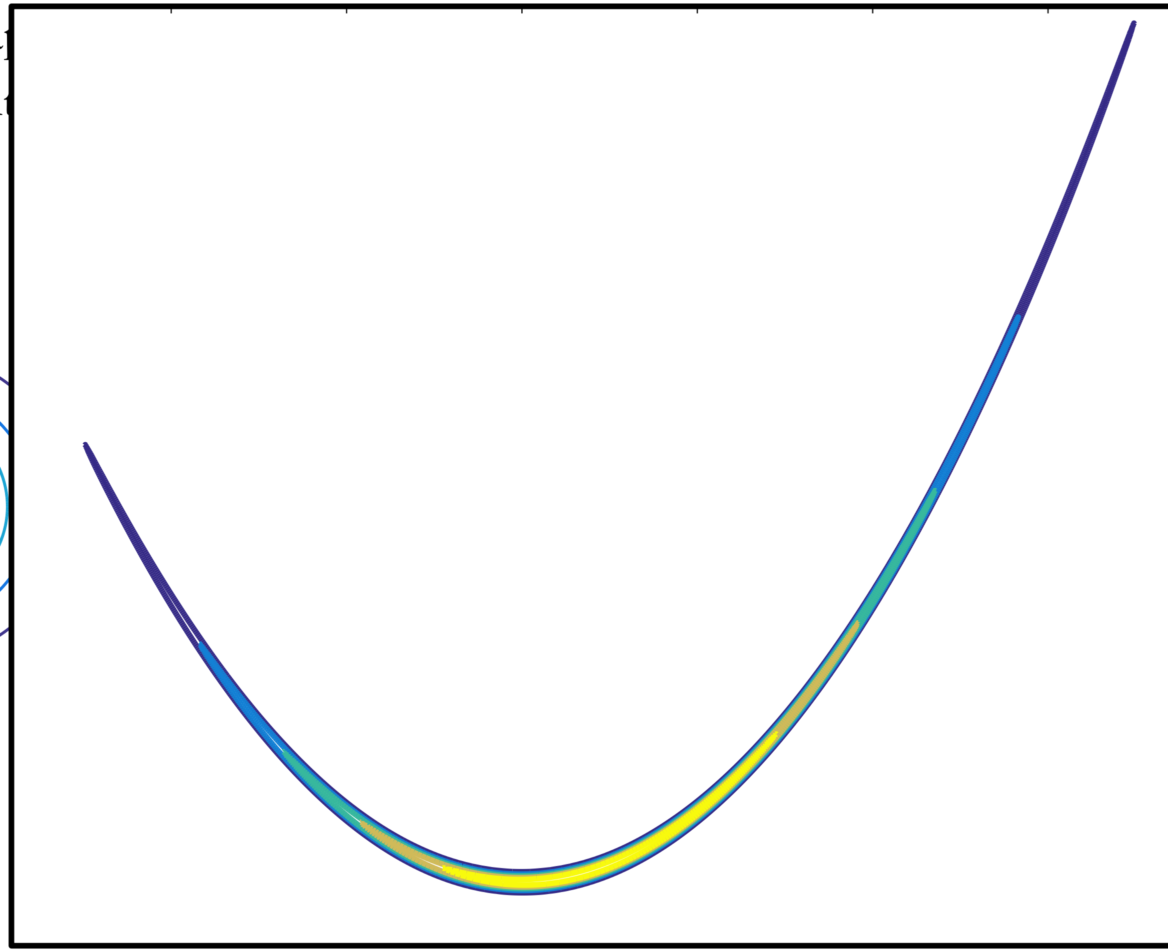
Convention holds that subset simulation is typically performed in a standard normal space

- For many a
- Component bly robust in this case

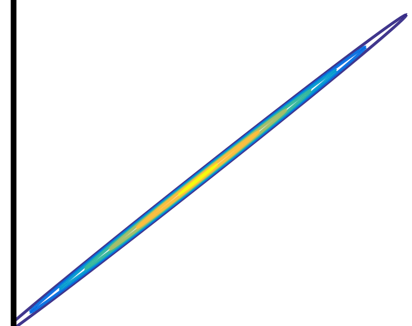
$\rho = 0$



MMH Good



$\rho = 0.999$



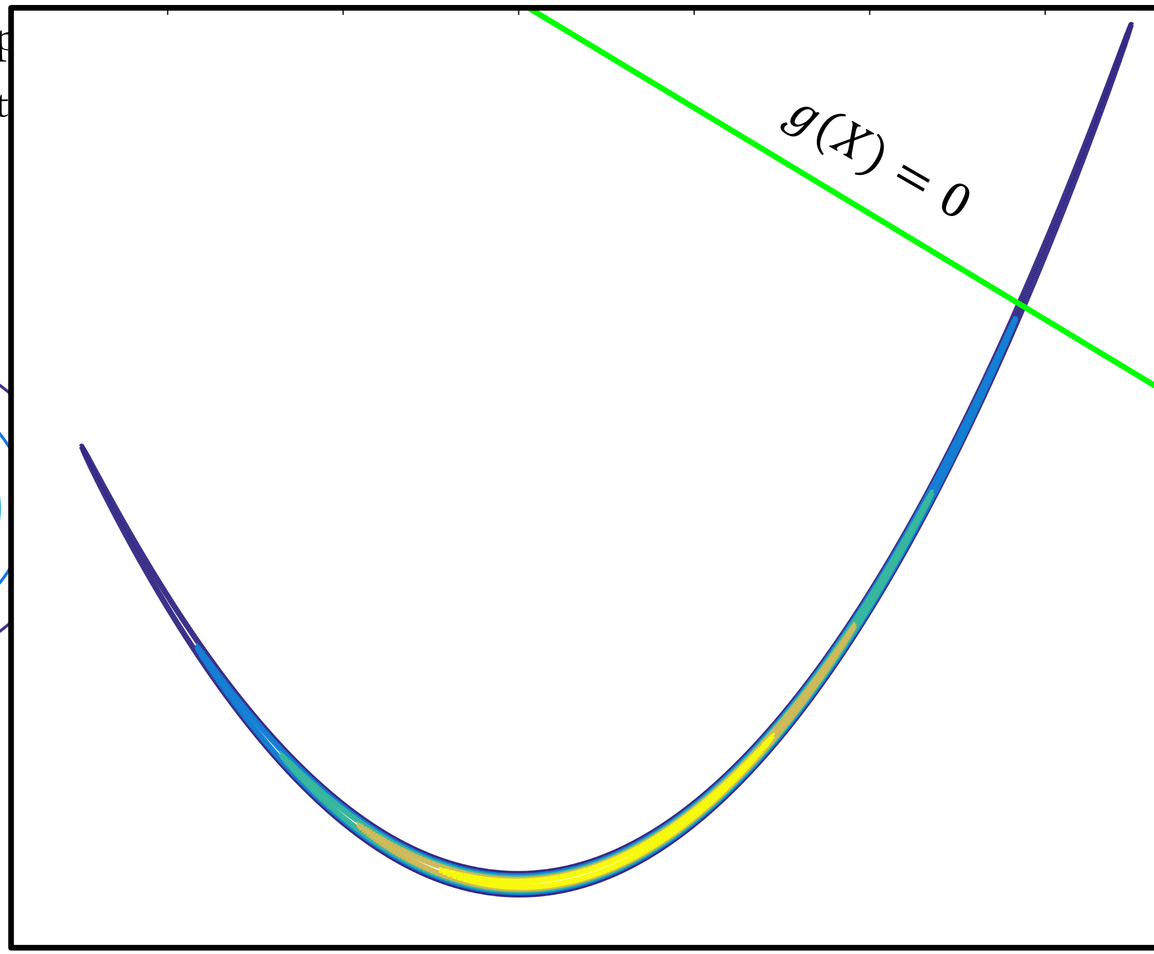
MMH Not Good

As the distribution degenerates, the correlation length of the MMH samples increases

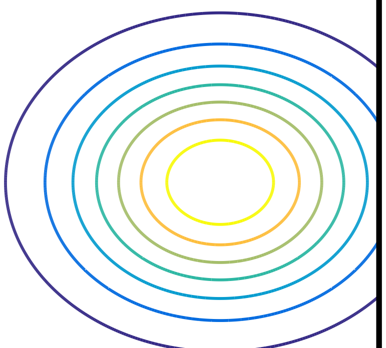
Degeneracy

Convention holds that subset simulation is typically performed in a standard normal space

- For many ap
- Component … ibly robust in this case

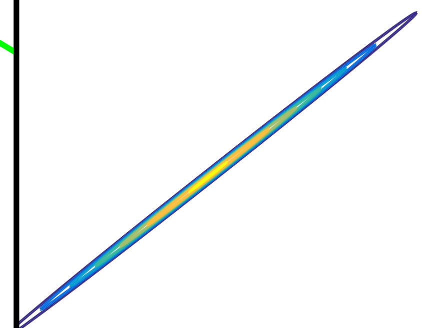


$\rho = 0$



MMH Good

$\rho = 0.999$



MMH Not Good

As the distribution degenerates, the correlation length of the MMH samples increases

Effect of Correlated Samples in Subset Simulation

- For Monte Carlo failure probability estimates from N independent samples, the coefficient of variation of the estimator scales as:

$$c = \sqrt{\frac{1-P_f}{P_f N}}$$

- In subset simulation, each conditional probability follows this relation with a correction for sample correlation as[1]:

$$c_i = \sqrt{\frac{1-P_f}{P_f N}(1+\gamma_i)}$$

where

$$\gamma_i = 2\sum_{\kappa=1}^{\frac{N}{N_c}-1}\left(1-\frac{kN_c}{N}\right)\rho_i(\kappa)$$

and $\rho_i(\kappa)$ is the correlation coefficient at lag κ.

The coefficient of variation over m conditional levels is given by:

$$c^2 = \sum_{i=1}^{m} c_i^2$$

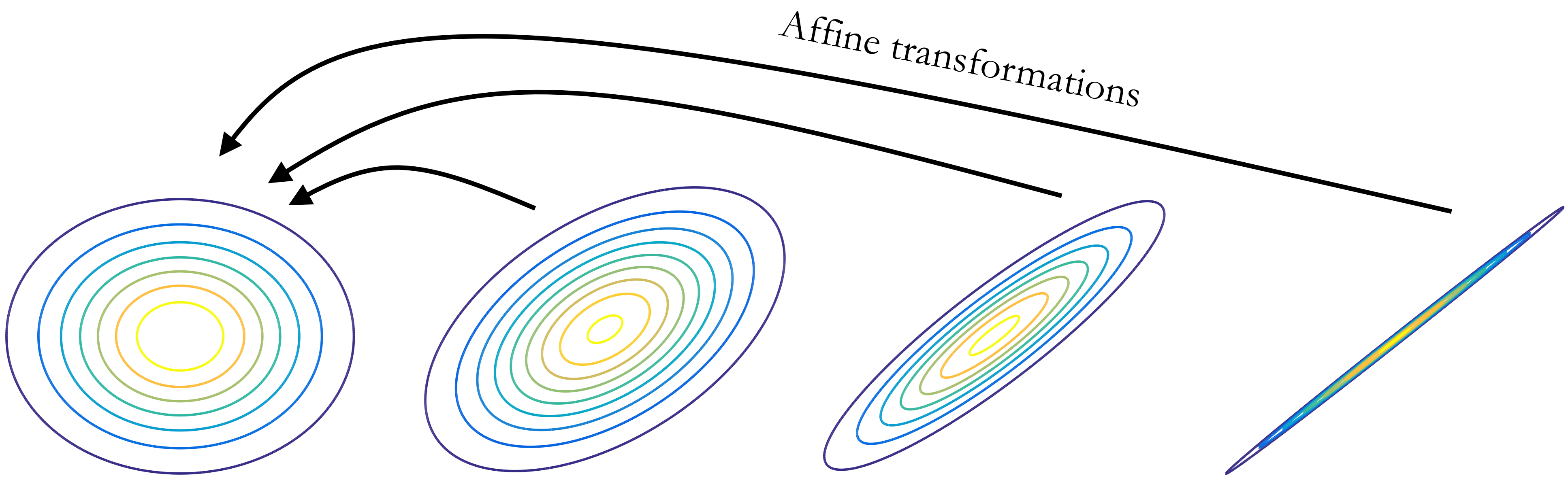
SURG Shields Uncertainty Research Group

[1]Au, S. and Beck, J. (2001). *Probabilistic Engineering Mechanics*. 16: 263-277

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

Affine Invariance

An affine transformation is an invertible mapping of the form $y = Ax + b$



Let $\pi_Y(y) = \pi_X(Ax + b) \propto \pi_X(x)$

A sampler is referred to as affine invariant if two sequences $\boldsymbol{X} = \{X_0, X_1, \dots, X_n\} \sim \pi_X(x)$ and $\boldsymbol{Y} = \{Y_0, Y_1, \dots, Y_n\} \sim \pi_Y(y)$ generated from the same random numbers can be related by[1]:

$$Y_i = AX_i + b$$

There are no known samplers with this property for any general class of densities[1]

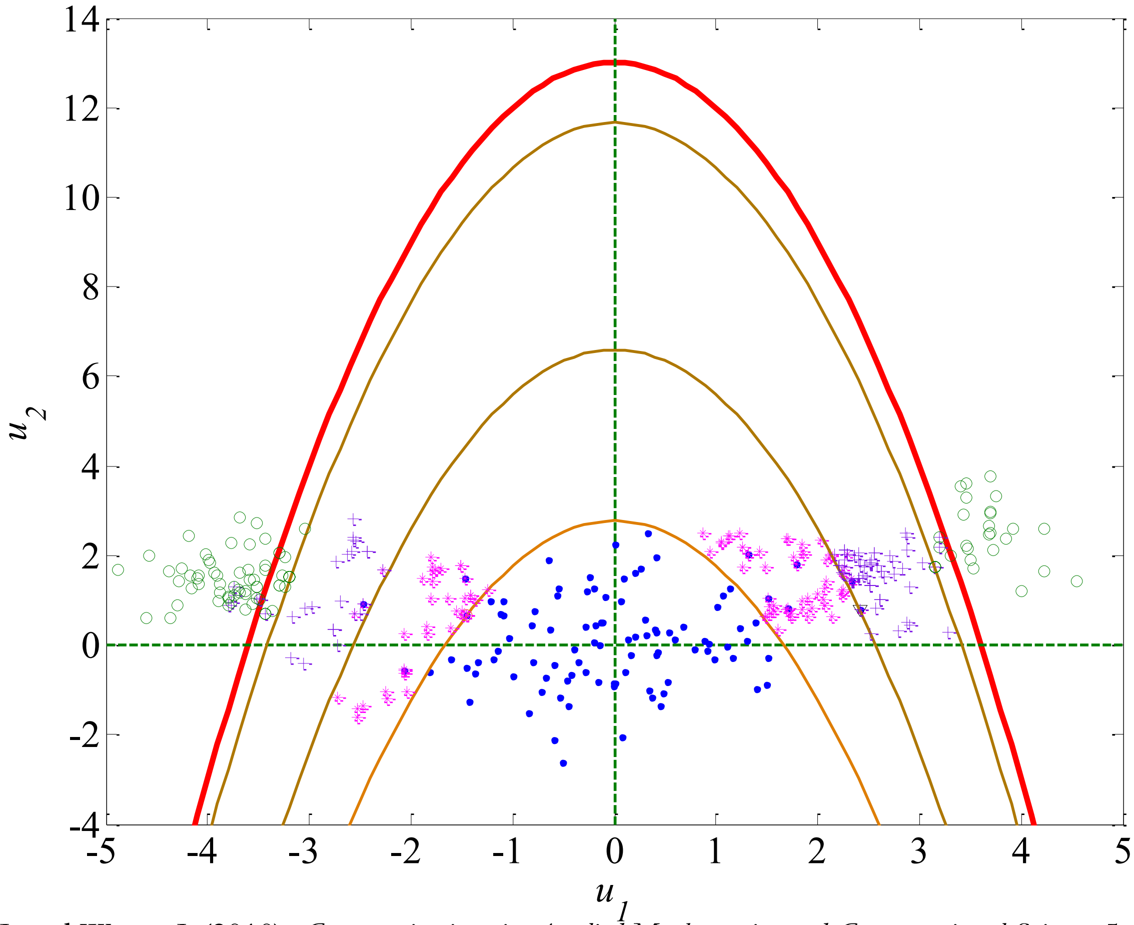
[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

Ensemble Samplers

An ensemble sampler is one that propagates a set of N_c independent sequences, each sampled from $\pi_X(x)$ that preserves the product density

$$\Pi(\boldsymbol{x}) = \pi_X\left(x^{(1)}\right)\pi_X\left(x^{(2)}\right)\cdots\pi_X\left(x^{(N_c)}\right)$$

The individual sequences do not need to be independent or even Markovian.[1]



[1] Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science.* 5: 65-80

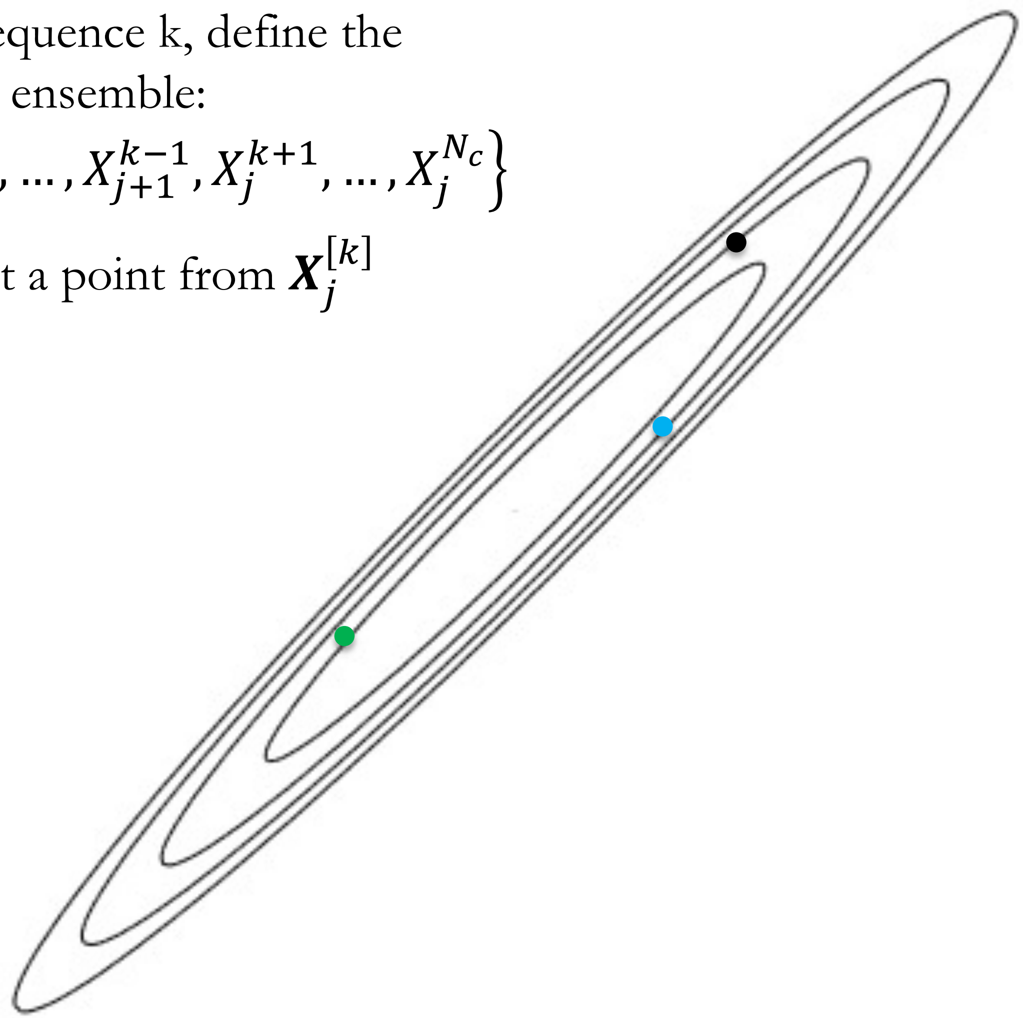
Affine Invariant Ensemble Sampler – Stretch Moves

Consider an ensemble of independent sequences

- To propagate sequence k, define the complementary ensemble:

  $$\boldsymbol{X}_j^{[k]} = \left\{X_{j+1}^1, \dots, X_{j+1}^{k-1}, X_j^{k+1}, \dots, X_j^{N_c}\right\}$$

- Randomly select a point from $\boldsymbol{X}_j^{[k]}$

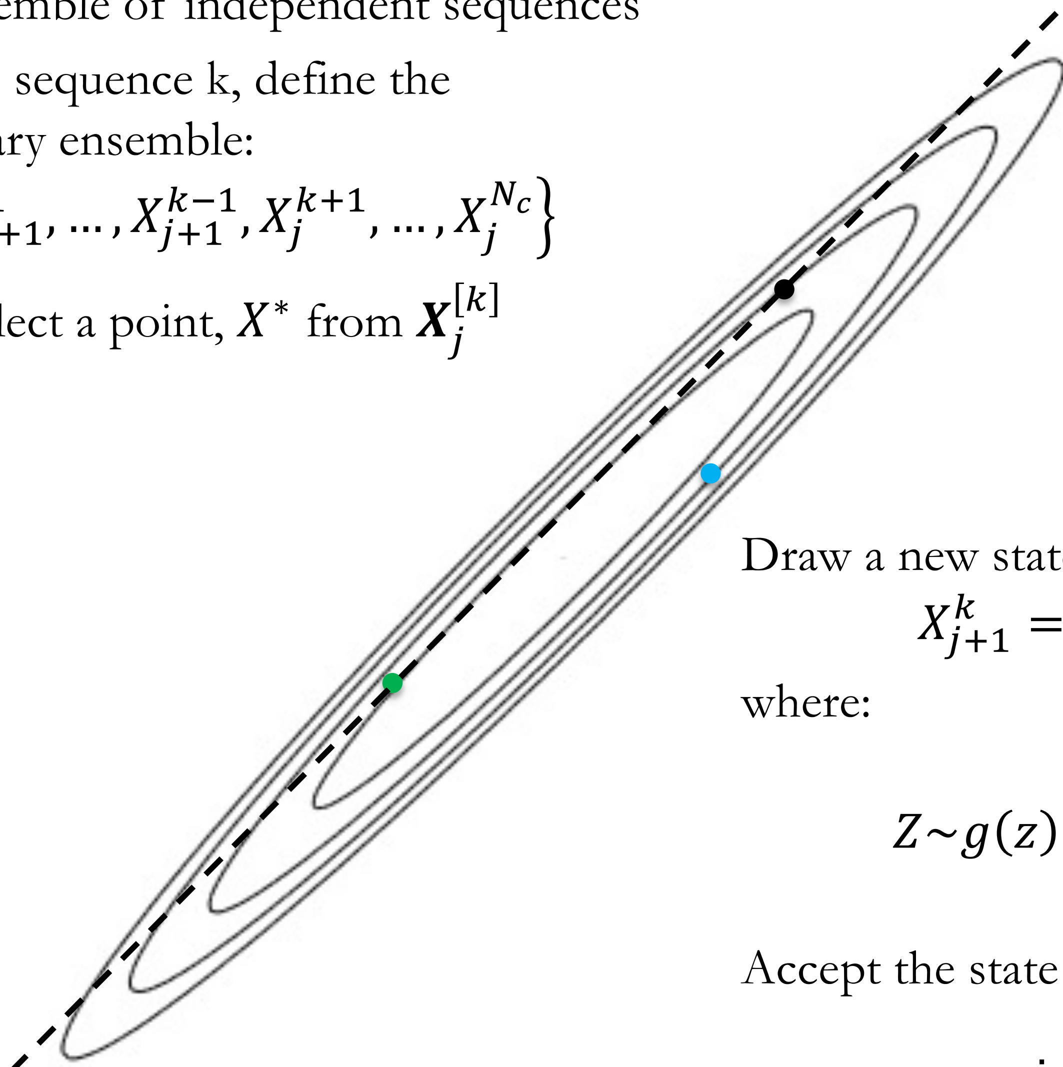


[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

Affine Invariant Ensemble Sampler – Stretch Moves

Consider an ensemble of independent sequences

- To propagate sequence k, define the complementary ensemble:
  $$\boldsymbol{X}_j^{[k]} = \left\{X_{j+1}^1, \ldots, X_{j+1}^{k-1}, X_j^{k+1}, \ldots, X_j^{N_c}\right\}$$
- Randomly select a point, X^* from $\boldsymbol{X}_j^{[k]}$



Draw a new state according to:

$$X_{j+1}^k = X^* + Z\left(X_j^k - X^*\right)$$

where:

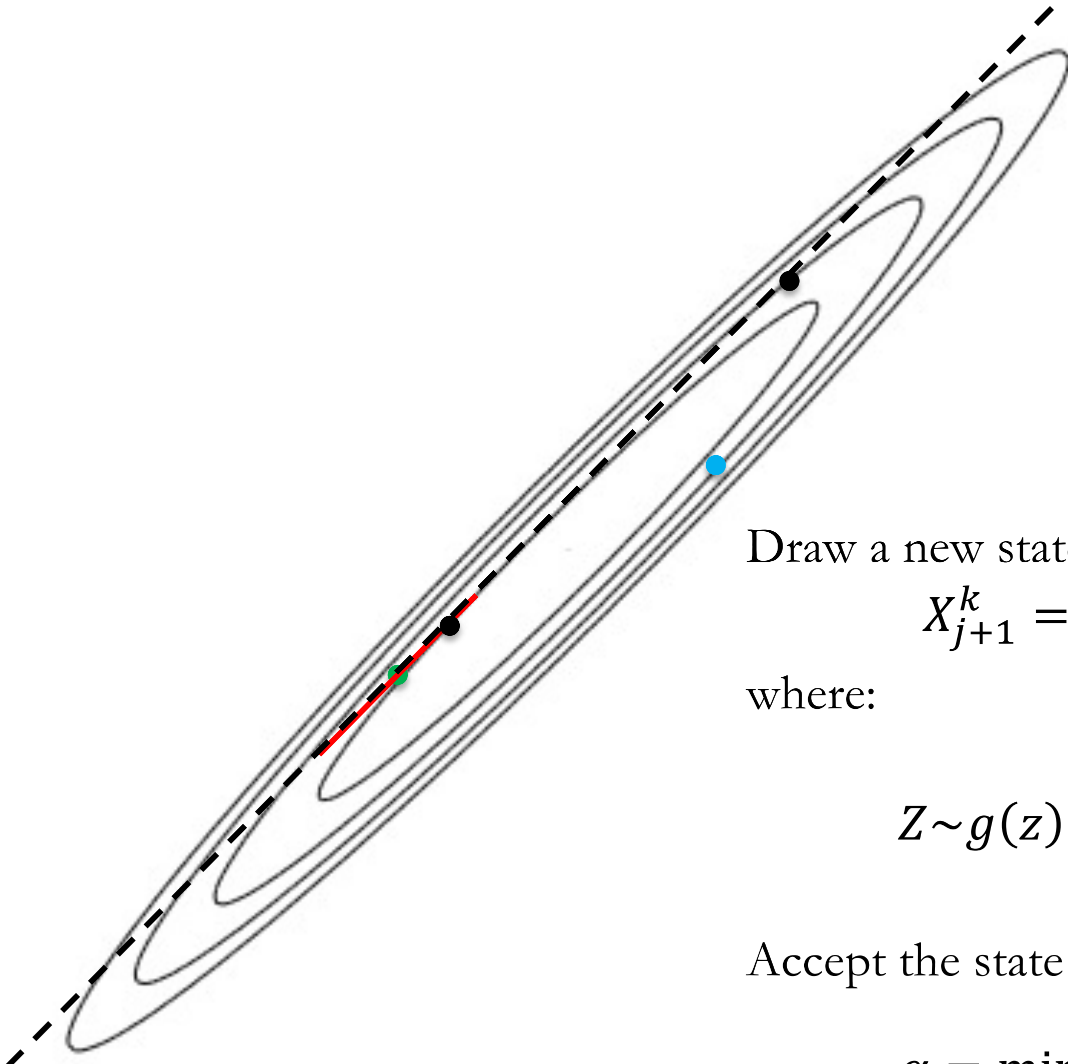
$$Z \sim g(z) \propto \begin{cases} \frac{1}{\sqrt{z}} & z \in \left[\frac{1}{a}, a\right] \\ 0 & o.w. \end{cases}$$

Accept the state with probability

$$\alpha = \min\left(1, z^{n-1}\frac{\pi\left(X_{j+1}^k\right)}{\pi\left(X_j^k\right)}\right)$$

[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science.* 5: 65-80

Affine Invariant Ensemble Sampler – Stretch Moves



Draw a new state according to:

$$X_{j+1}^k = X^* + Z\left(X_j^k - X^*\right)$$

where:

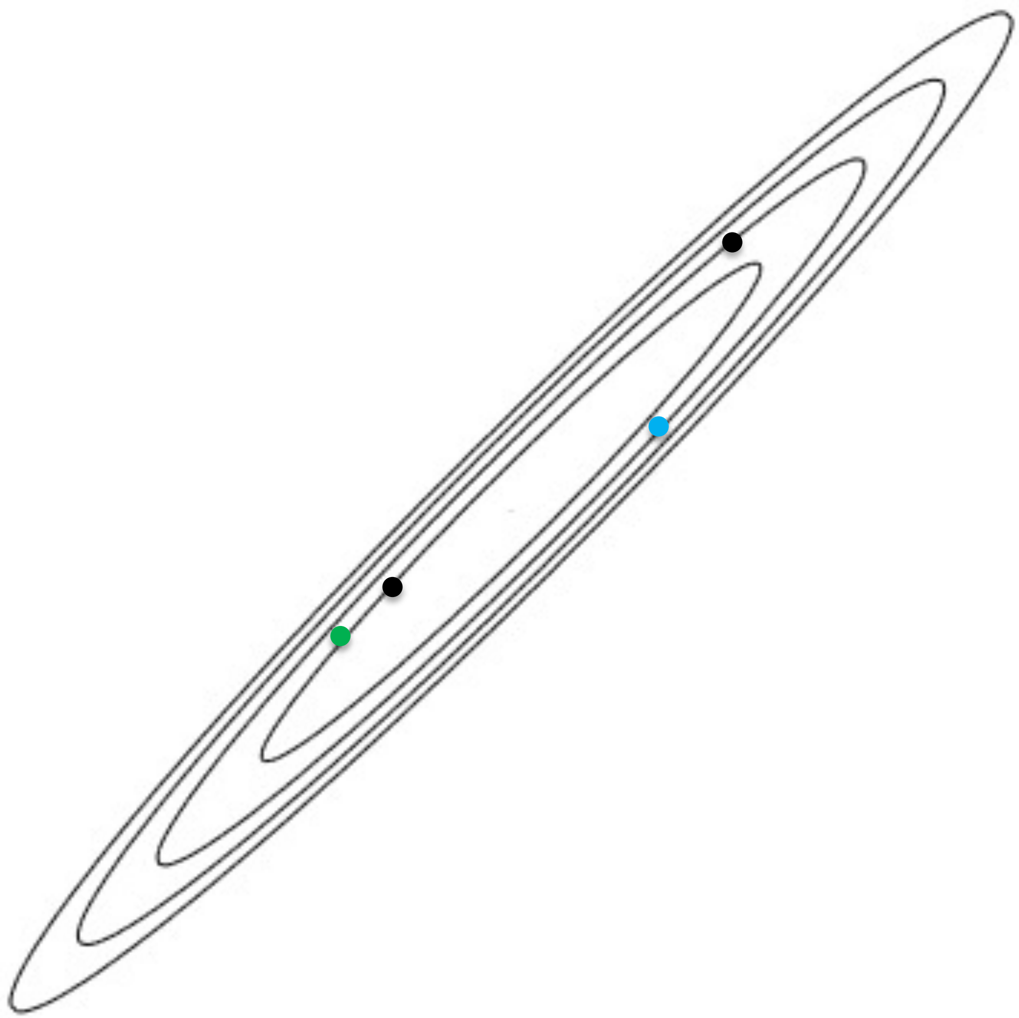
$$Z \sim g(z) \propto \begin{cases} \frac{1}{\sqrt{z}} & z \in \left[\frac{1}{a}, a\right] \\ 0 & o.w. \end{cases}$$

Accept the state with probability

$$\alpha = \min\left(1, z^{n-1} \frac{\pi\left(X_{j+1}^k\right)}{\pi\left(X_j^k\right)}\right)$$

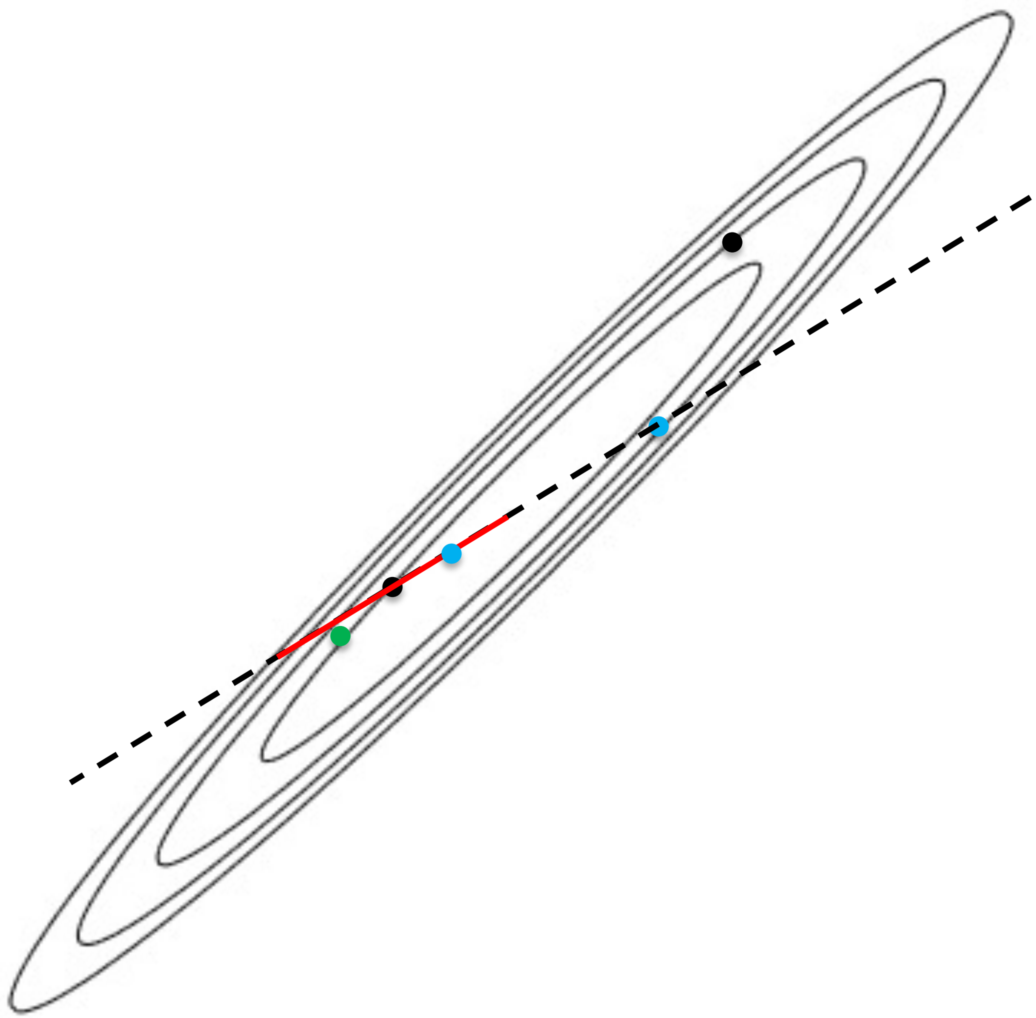
[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

Affine Invariant Ensemble Sampler – Stretch Moves



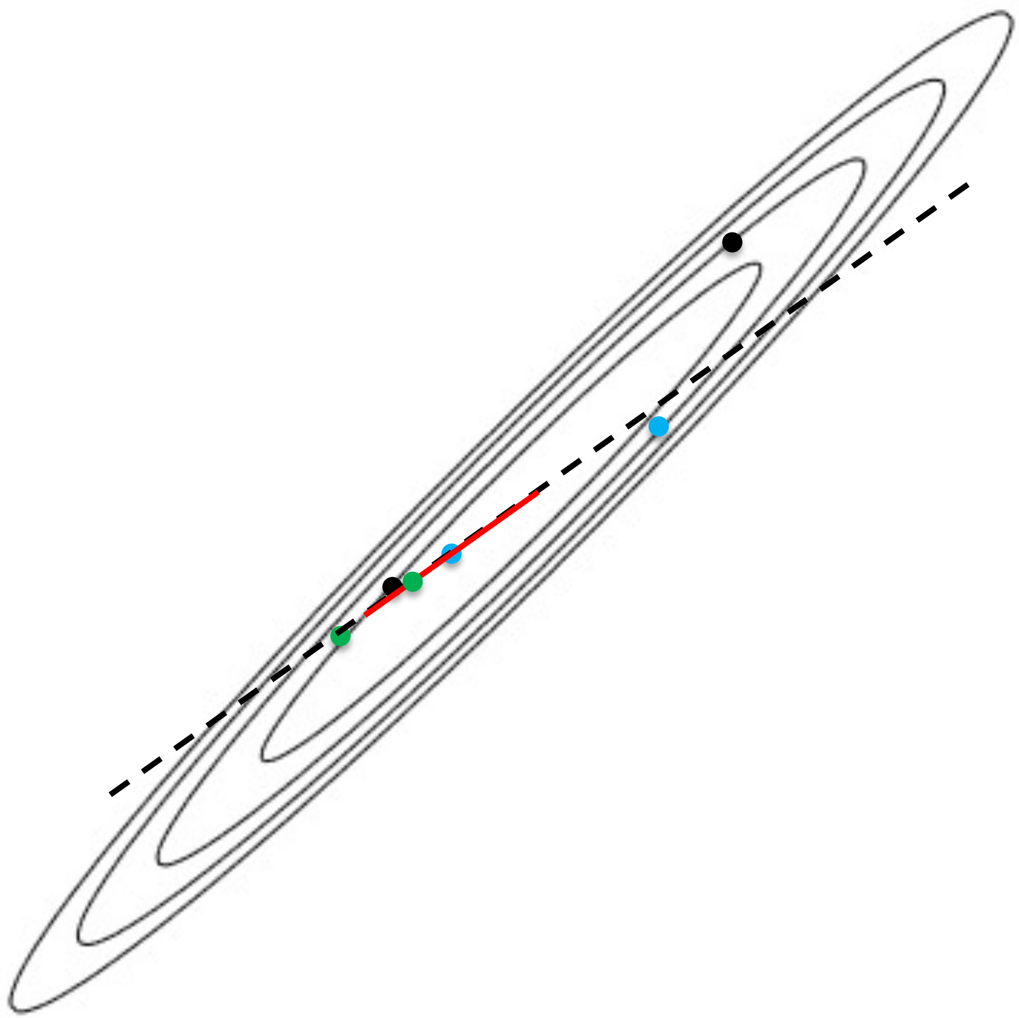
[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science.* 5: 65-80

Affine Invariant Ensemble Sampler – Stretch Moves



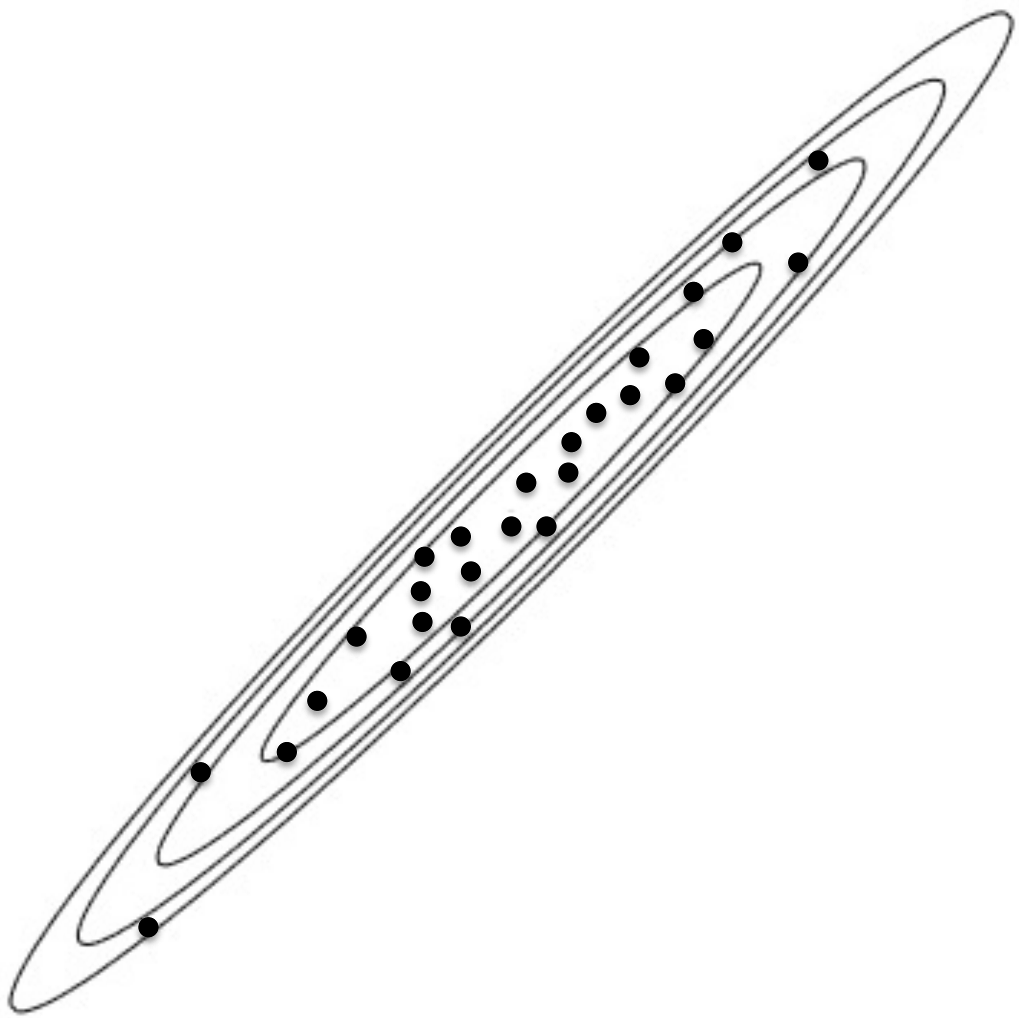
[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science.* 5: 65-80

Affine Invariant Ensemble Sampler – Stretch Moves



[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

Affine Invariant Ensemble Sampler – Stretch Moves



SURG Shields Uncertainty Research Group

[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

Johns Hopkins Whiting School of Engineering

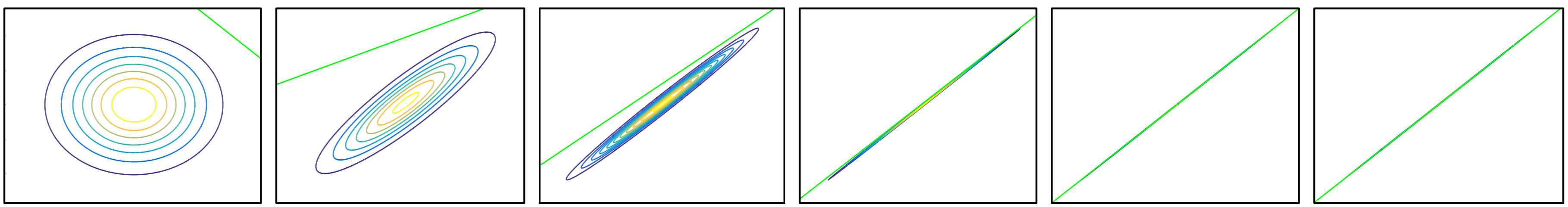
Example 1: Degenerate Gaussian

Performance Function in Standard Normal Space:

$$g(U) = \beta\sqrt{n} + \sum_{i=1}^{n} U_i$$

$\boldsymbol{U} =$ n-dimensional standard normal random vector

$\beta = 3, P_F = 1.3\times10^{-3}$



$n\backslash\rho$		0	0.9	0.99	0.9999	0.999999	0.99999999
2	MMH	1.36e-3 (39%)	1.83e-3 (46%)	1.43e-3 (68%)	1.08e-3 (128%)	1.12e-3 (132%)	1.10e-3 (130%)
	AIE	1.52e-3 (63%)	1.42e-3 (74%)	1.29e-3 (83%)	1.38e-3 (75%)	1.36e-3 (64%)	1.44e-3 (63%)
10	MMH	1.35e-3 (39%)	1.37e-3 (39%)	1.46e-3 (43%)	1.31e-3 (92%)	1.46e-3 (115%)	1.17e-3 (129%)
	AIE	1.42e-3 (87%)	1.42e-3 (92%)	1.34e-3 (91%)	1.37e-3 (79%)	1.31e-3 (92%)	1.50e-3 (80%)
50	MMH	1.43e-3 (35%)	1.36e-3 (33%)	1.39e-3 (32%)	1.40e-3 (82%)	1.39e-3 (96%)	1.46e-3 (126%)
	AIE	1.45e-3 (95%)	1.26e-3 (123%)	1.23e-3 (123%)	1.42e-3 (100%)	1.49e-3 (98%)	1.45e-3 (103%)

Example 2: Rosenbrock Distribution

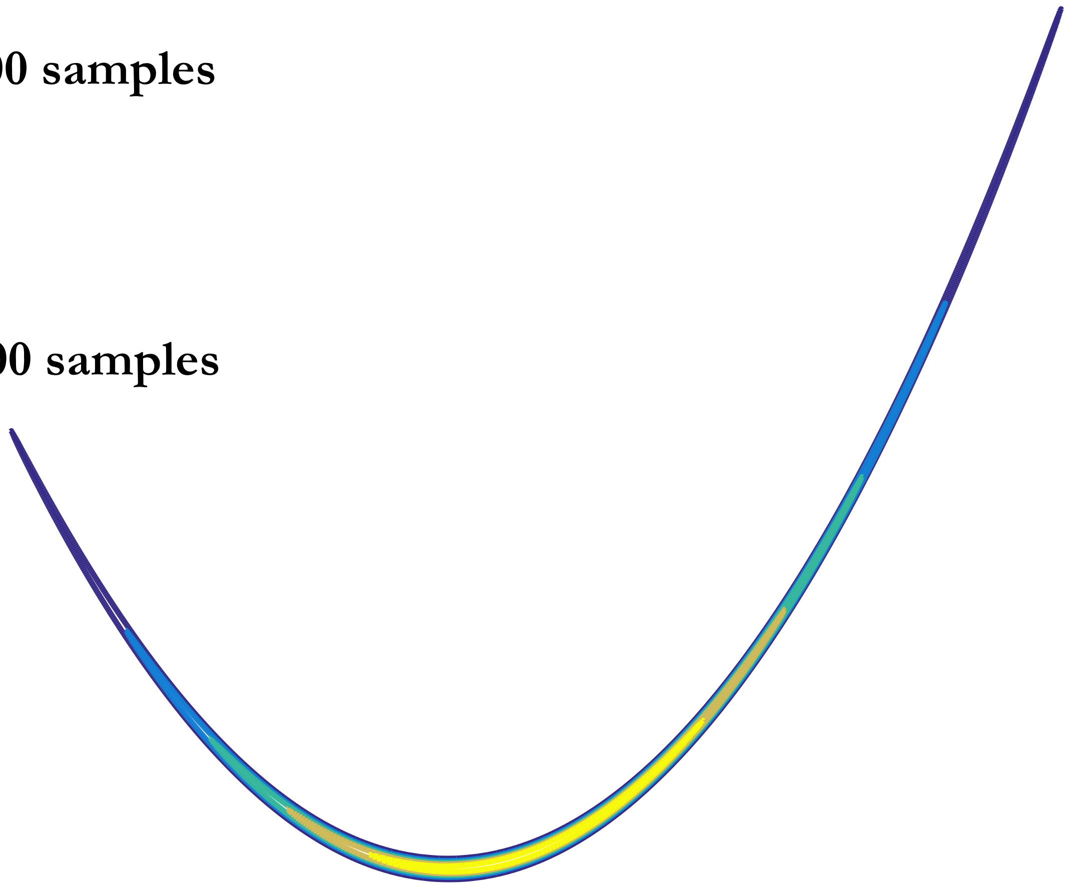
$$\pi(\boldsymbol{x}) \propto \exp\left(-\frac{100(x_2 - x_1^2)^2 + (1 - x_1)^2}{20}\right)$$

[1]Reported Correlation Lengths, x_1

- Modified Metropolis-Hastings: **163,000 samples**
- Stretch Moves: **19,400 samples**

[1]Reported Correlation Lengths, x_2

- Modified Metropolis-Hastings: **322,000 samples**
- Stretch Moves: **67,000 samples**



[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science.* 5: 65-80

Example 2: Rosenbrock Distribution

$$\pi(\boldsymbol{x}) \propto \exp\left(-\frac{100(x_2 - x_1^2)^2 + (1 - x_1)^2}{20}\right)$$

[1]Reported Correlation Lengths, x_1

- Modified Metropolis-Hastings: **163,000 samples**
- Stretch Moves: **19,400 samples**

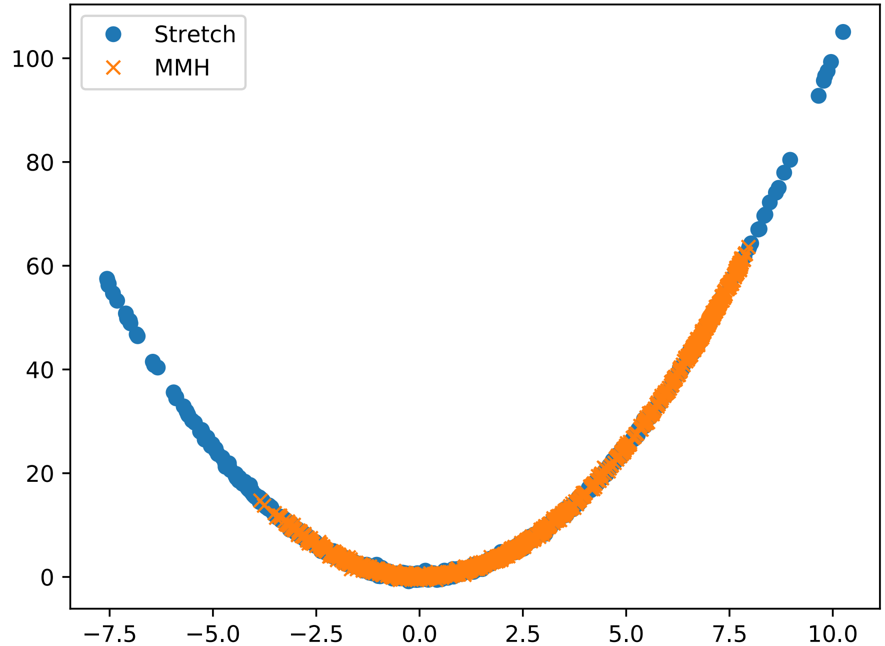
[1]Reported Correlation Lengths, x_2

- Modified Metropolis-Hastings: **322,000 samples**
- Stretch Moves: **67,000 samples**

$N = 1{,}000$ samples

Jumping $N_j = 1{,}000$ samples

$1{,}000{,}000$ total samples

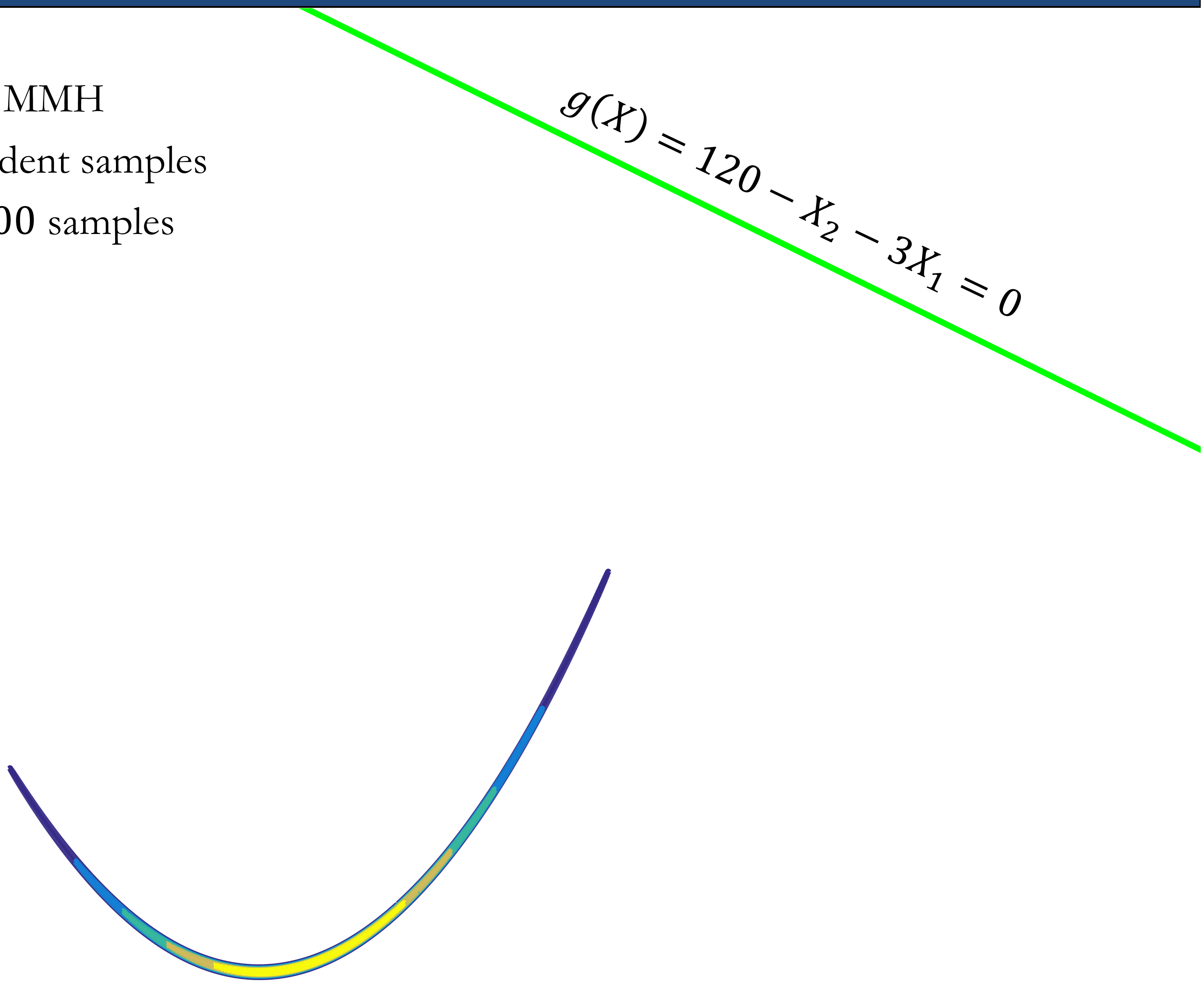


[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science.* 5: 65-80

Example 2: Rosenbrock Distribution

Monte Carlo Simulation: MMH

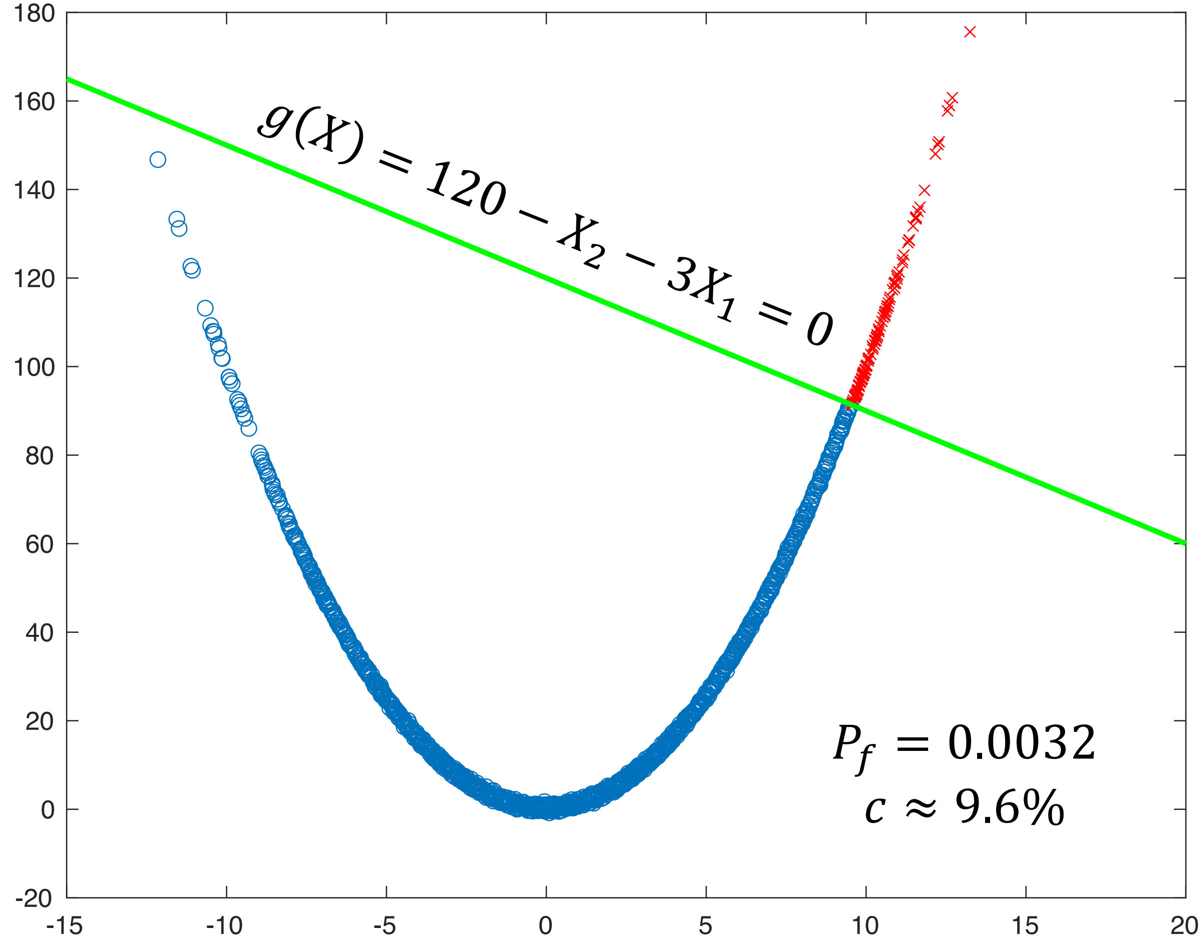
- $N = 33{,}333$ independent samples
- Jumping $N_j = 300{,}000$ samples
- 10^{10} total samples



Example 2: Rosenbrock Distribution

Monte Carlo Simulation: MMH

- $N = 33{,}333$ independent samples
- Jumping $N_j = 300{,}000$ samples
- 10^{10} total samples



SURG Shields Uncertainty Research Group

Johns Hopkins Whiting School of Engineering

Example 2: Rosenbrock Distribution

Monte Carlo Simulation

c	N	N_{MMH}	$N_{stretch}$
0.1	31,150	1.003×10^{10}	2.087×10^{9}
0.25	4,984	1.6048×10^{9}	3.3393×10^{8}
0.5	1,246	4.0121×10^{8}	8.3482×10^{7}

Subset Simulation

c	N	N_{MMH}	$N_{stretch}$
0.1	6,037	1.9439×10^{9}	4.0448×10^{8}
0.25	966	3.1105×10^{8}	6.4722×10^{7}
0.5	241	7.7602×10^{7}	1.6147×10^{7}

Example 3: Stochastic PDE

$$\pi(\boldsymbol{u}) \propto \exp\left(-\int_0^1 \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 + V\big(u(x)\big)dx\right)$$

where

$$V(u) = (1-u^2)^2$$

$\pi(\boldsymbol{u})$ is the invariant distribution of the stochastic Allen-Cahn equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - V'(u) + \sqrt{2}\eta$$

Discretizing $\pi(\boldsymbol{u})$ gives

$$\pi(\boldsymbol{u}) = exp\left(-\sum_{i=0}^{d-1} \frac{1}{2\Delta x}\Big(u\big((i+1)\Delta x\big) - u(i\Delta x)\Big)^2 + \frac{\Delta x}{2}\Big(V\Big(u\big((i+1)\Delta x\big)\Big) + V\big(u(i\Delta x)\big)\Big)\right)$$

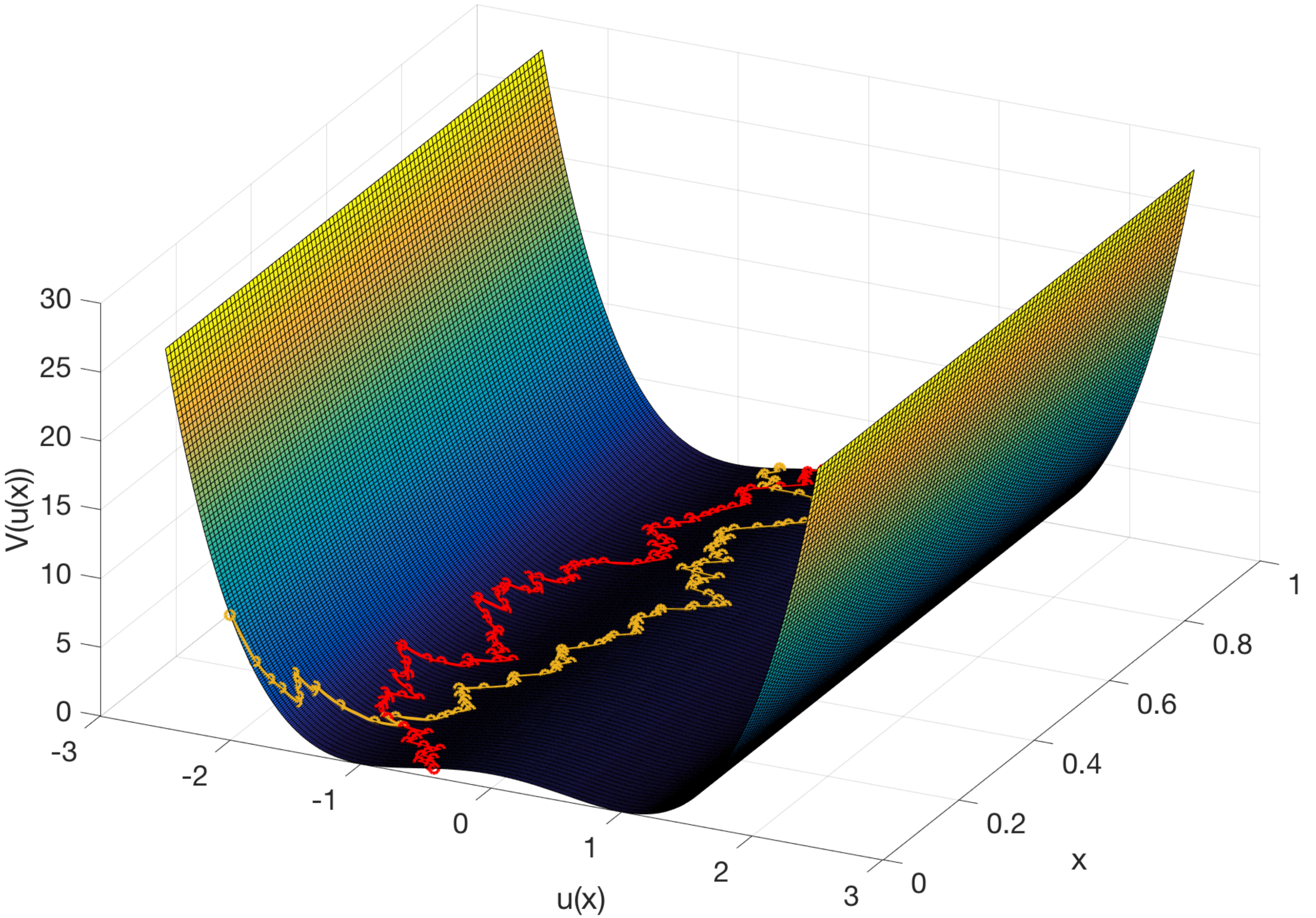
where $\Delta x = \frac{1}{d}$, $d = 100$.

[1]Reported Correlation Lengths, $\int_0^1 u(x)dx$

- Modified Metropolis-Hastings: **80,000 samples**
- Stretch Moves: **5,200 samples**

[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

Example 3: Stochastic PDE



Example 3: Stochastic PDE – MCS

Define performance function: $g(u) = \min(u(x))$

Define limit surface: $g(u) + 2.5 = 0$

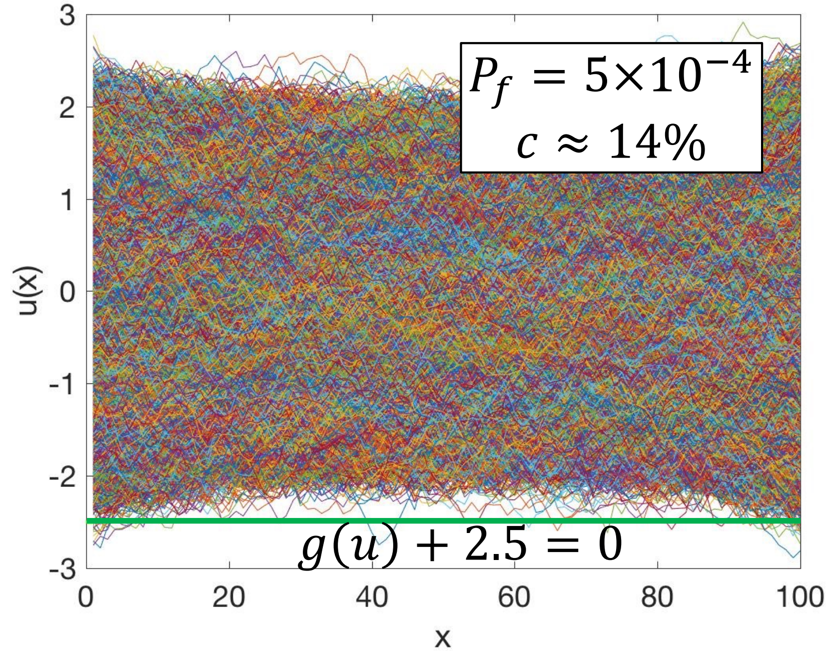
Monte Carlo Simulation: (100,000 ind. samples)

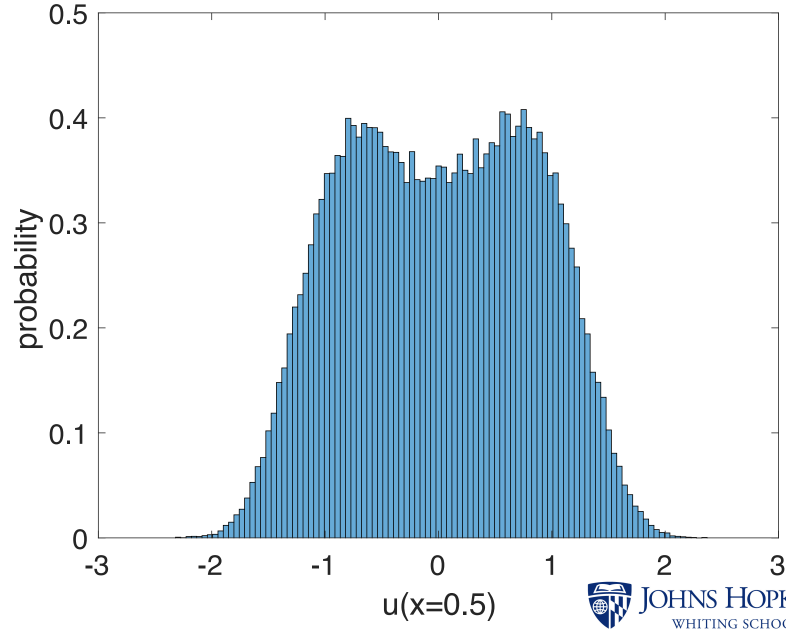
Modified Metropolis-Hastings: Requires $\sim 10^{10}$ samples

- 36 hours x 70 CPUs

Affine Invariant – Stretch: Requires 6×10^{8} samples

- 18 hours x 1 CPU
- Saves 9.4 billion samples!





Example 3: Stochastic PDE

Monte Carlo Simulation

c	N	N_{MMH}	$N_{stretch}$
0.1	199,900	1.5992×10^{10}	1.0394×10^{9}
0.25	31,984	2.5587×10^{9}	1.6631×10^{8}
0.5	7,996	6.3968×10^{8}	4.1579×10^{7}

Subset Simulation

c	N	N_{MMH}	$N_{stretch}$
0.1	9,138	7.3104×10^{8}	4.7517×10^{7}
0.25	1462	1.1696×10^{8}	7.6024×10^{6}
0.5	365	2.92×10^{8}	1.898×10^{6}

[1]Goodman, J. and Weare, J. (2010). *Communications in Applied Mathematics and Computational Science*. 5: 65-80

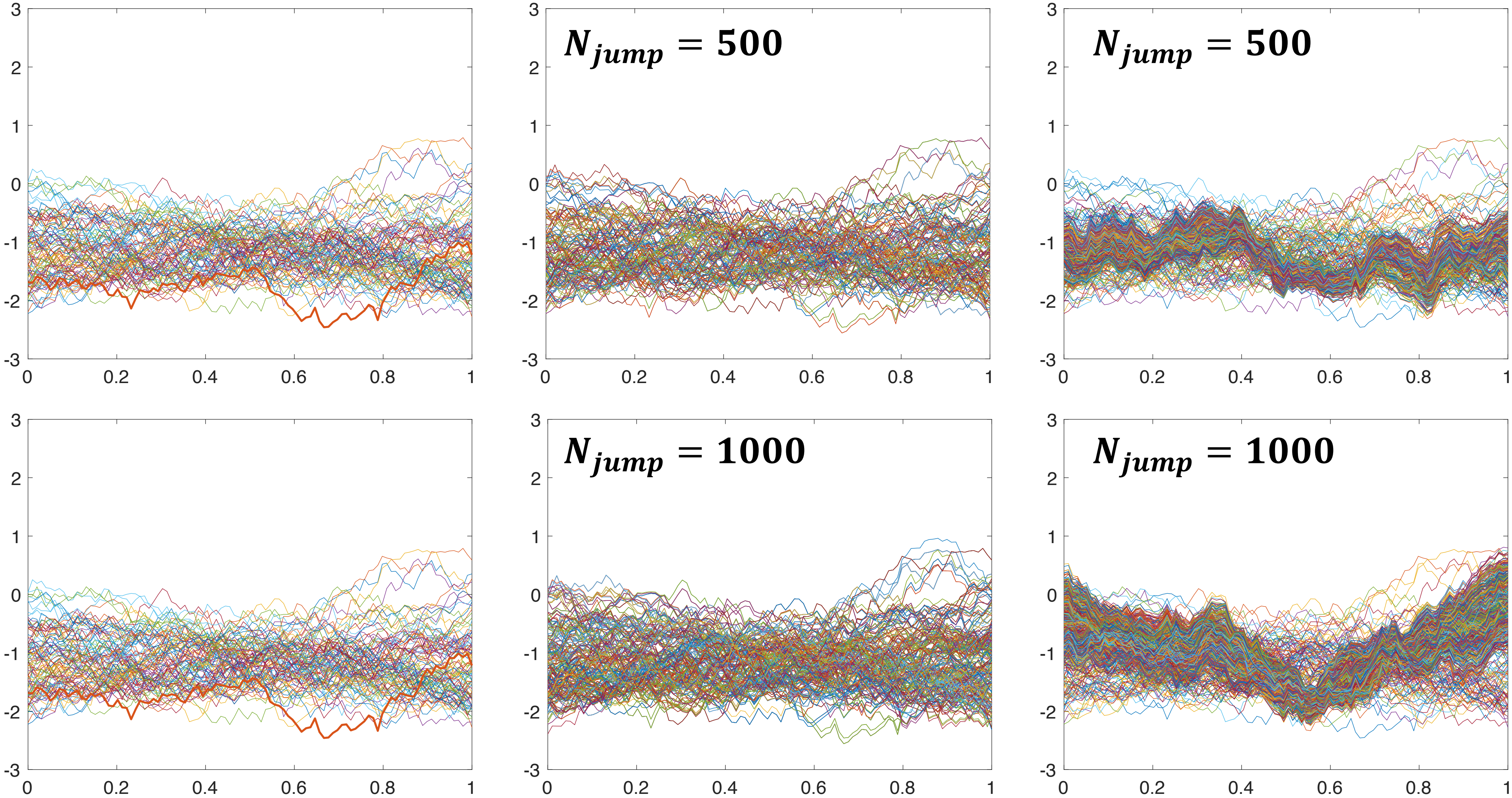
Example 3: Stochastic PDE

S U R G
Shields Uncertainty Research Group

Johns Hopkins
Whiting School of Engineering

Example 3: Stochastic PDE

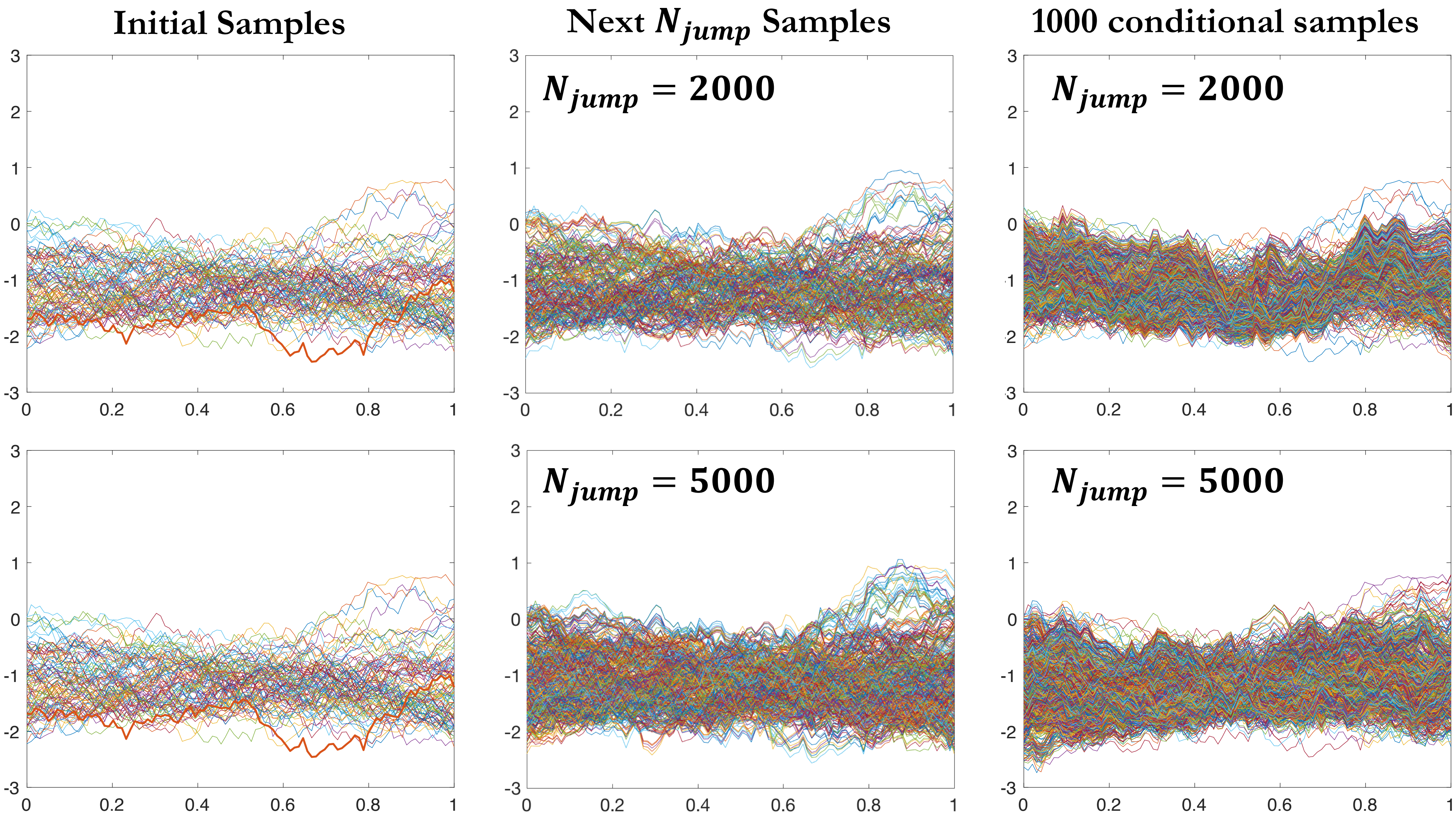
Initial Samples | Next N_{jump} Samples | 1000 conditional samples



SURG
Shields Uncertainty Research Group

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Example 3: Stochastic PDE

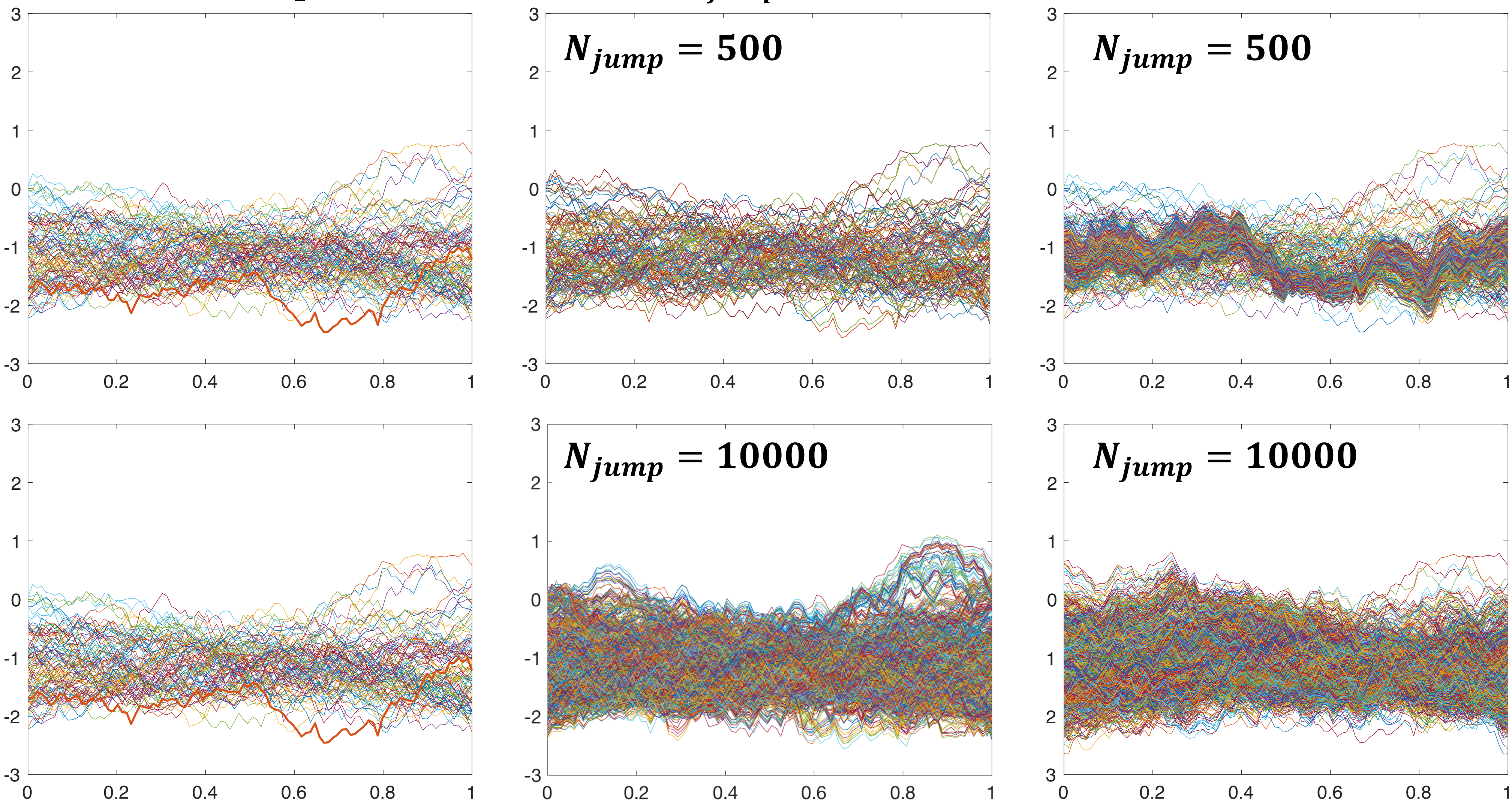


SURG Shields Uncertainty Research Group

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

Example 3: Stochastic PDE

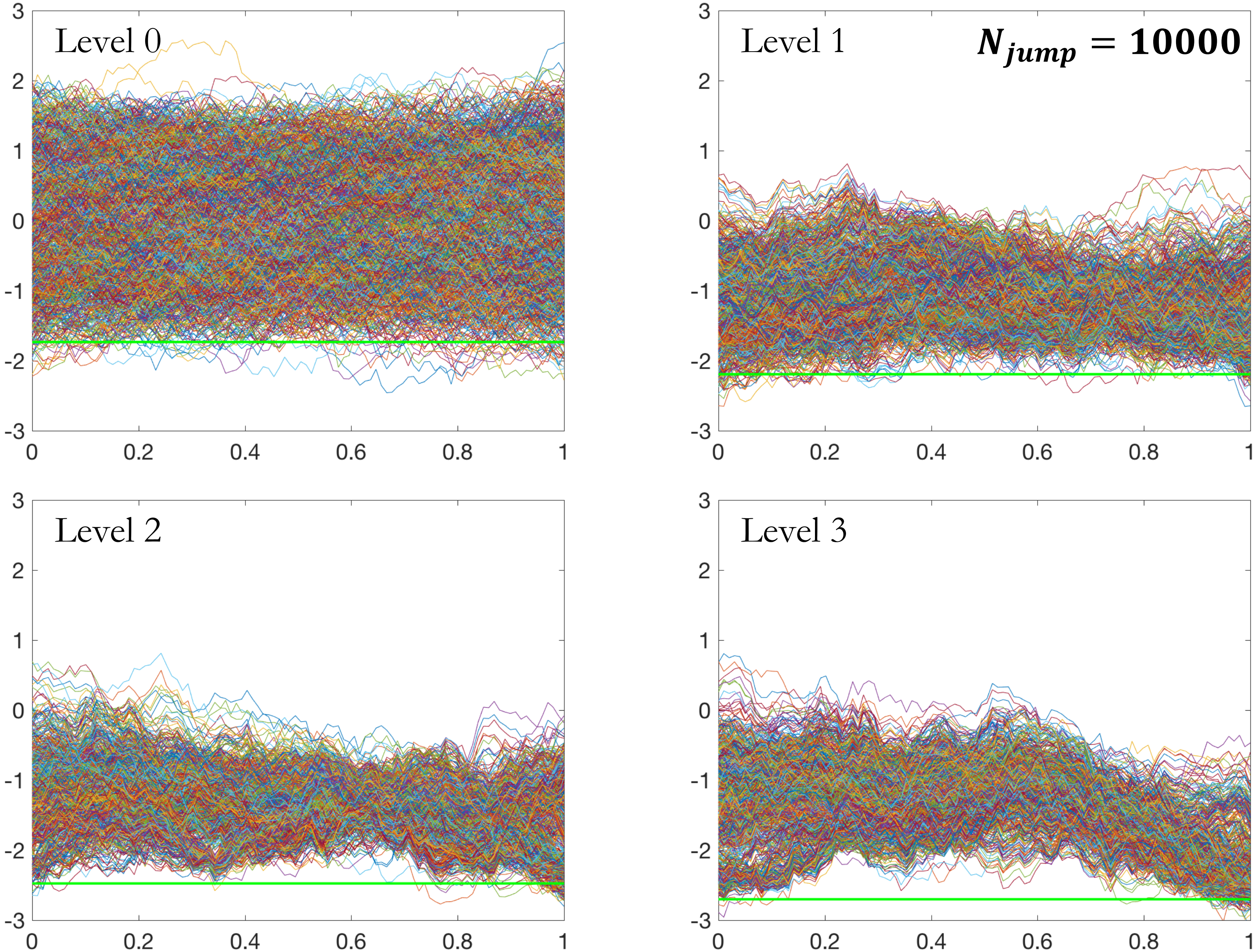
Initial Samples | Next N_{jump} Samples | 1000 conditional samples



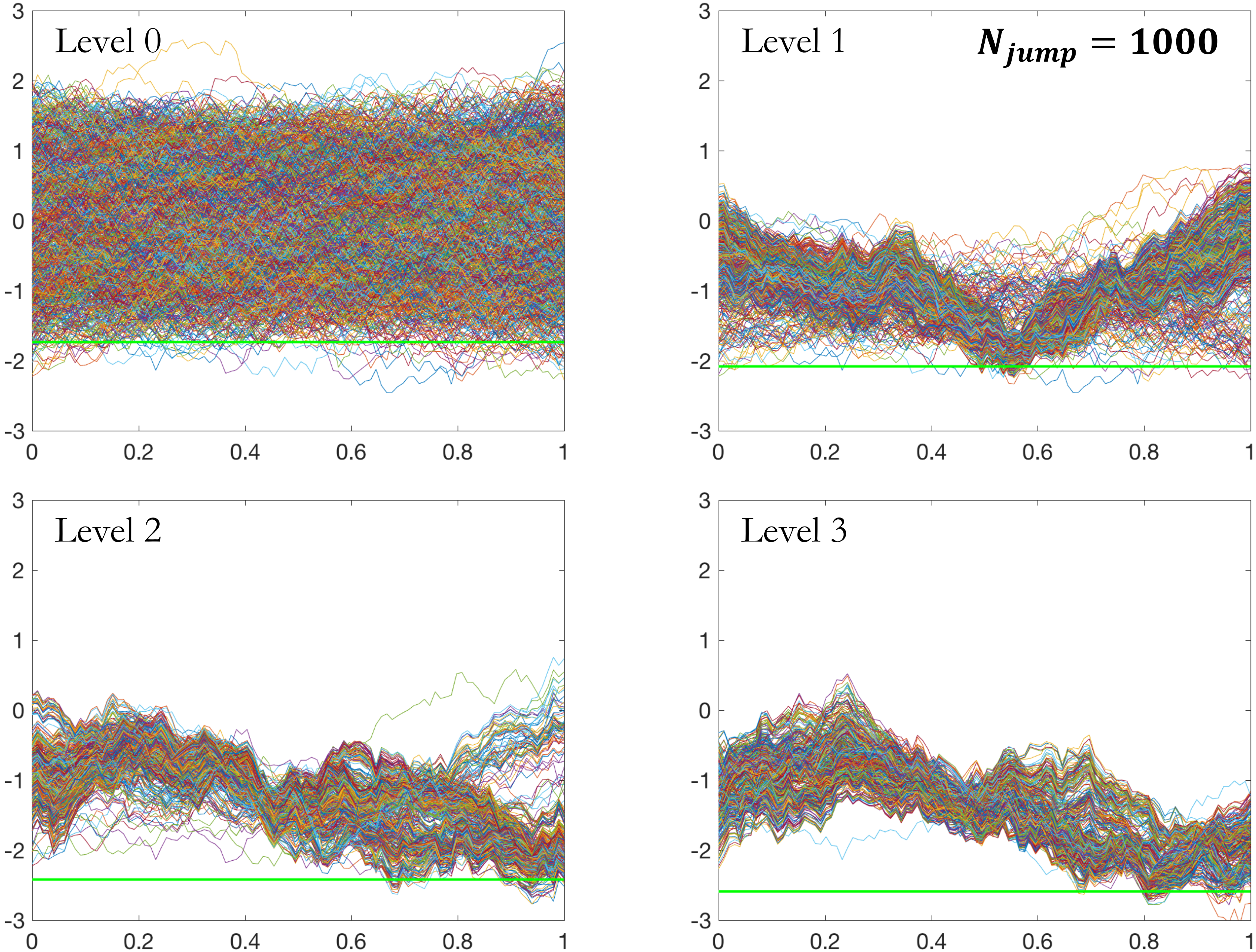
SURG Shields Uncertainty Research Group

Johns Hopkins Whiting School of Engineering

Example 3: Stochastic PDE



Example 3: Stochastic PDE



Example 3: Stochastic PDE – Subset Simulation

Define performance function: $g(u) = \min(u(x))$

Define limit surface: $g(u) + 2.5 = 0$

Subset Simulation: (1,000 samples per conditional level, 50 ind. trials)

Affine Invariant – Stretch Moves

N_{jump}	N	$\overline{P}_f$	c
10,000	4×10^7	4.919×10^{-4}	0.45
5,000	2×10^7	5.347×10^{-4}	0.54
2,000	8×10^6	5.699×10^{-4}	0.7
1,000	4×10^6	6.141×10^{-4}	1.28
500	2×10^6	6.655×10^{-4}	2.04

Summary

- Subset simulation has been investigated for cases where input distributions are non-Gaussian and poorly scaled.
- It is proposed to use the affine invariant ensemble MCMC sampler to generate samples from each conditional level
  - These samples have much smaller correlation length
  - Reduces the number of samples by several orders of magnitude

- These problems are still very computationally expensive – requiring millions of samples even with the affine invariant sampler.

Acknowledgments

Sponsors

ONR Young Investigatory Award, Project No. N00014-15-1-2754

Thank You!

Questions?