NUMERICAL METHODS FOR STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS AND THEIR CONTROL

Max Gunzburger

Department of Scientific Computing, Florida State University

gunzburg@scs.fsu.edu

London Mathematical Society Durham Symposium

Computational Linear Algebra for Partial Differential Equations

July 14 – 24, 2008, Durham, UK

All models are wrong, but some models are useful

George Box

Computational results are believed by no one, except for the person who wrote the code

Experimental results are believed by everyone, except for the person who ran the experiment

NUMERICAL METHODS FOR STOCHASTIC PDE'S FOR DUMMIES

WHERE I AM THE DUMMY

INTRODUCTORY REMARKS

Uncertainty is everywhere

- Physical, biological, social, economic, financial, etc. processes always involve uncertainties

- As a result, mathematical models of these processes should account for uncertainty

- Accounting for uncertainty in processes governed by partial differential equations can involve

  – random coefficients in the PDE, boundary condition, and initial condition operators

  – random right-hand sides in the PDE's, boundary conditions, and initial conditions

  – random geometry, i.e., random boundary shapes

- Uncertainty arises because
  - available data are incomplete
    - they are predictable but are too difficult (perhaps impossible) or costly to obtain by measurement
      $\longrightarrow$ media properties in oil reservoirs or aquifers
    - they are unpredictable
      $\longrightarrow$ wind shear, rainfall amounts
  - not all scales in the data and/or solutions can or should be resolved
    - it is too difficult (perhaps impossible) or costly to do so in a computational simulation
      $\longrightarrow$ turbulence, molecular vibrations
    - some scales may not be of interest
      $\longrightarrow$ surface roughness, hourly stock prices

- Of course, it is well known that two experiments run under the "same" conditions will yield different results

Modeling noise

- White noise – input data vary randomly and independently from one point of the physical domain to another and from one time instant to another
  - uncertainty is described in terms of uncorrelated random fields
  - examples: thermal fluctuations; surface roughness; Langevin dynamics

- Colored noise – input data vary randomly from one point of the physical domain to another and from one time instant to another according to a given (spatial/temporal) correlation structure
  - uncertainty is described in terms of correlated random fields
  - examples: rainfall amounts; bone densities; permeabilities within subsurface layers

- Random parameters – input data depend on a finite[†] number of random parameters
  - think of this case as "knobs" in an experiment
  - each parameter may vary independently according to its own given probability density
  - alternately, the parameters may vary according to a given joint probability density
  - examples: homogeneous material properties, e.g., Young's modulus, Poisson's ratio, speed of sound, inflow mass

[†]What we really mean is that the number of parameters is not only finite, but independent of the spatial/temporal discretization; this is not possible for the approximation of white noise for which the number of parameters increases as the grid sizes decrease

- Ultimately, for all three cases, on a computer
  one solves problems involving random parameters

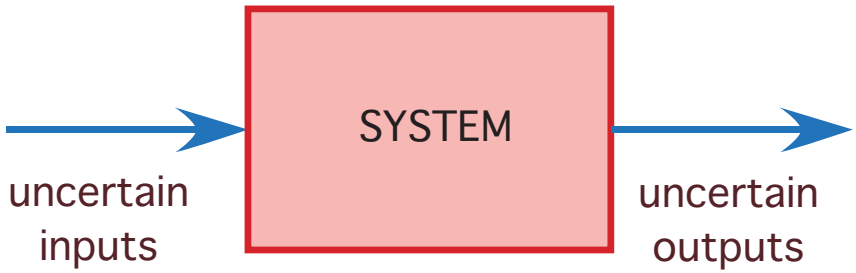
  – in the white noise and colored noise cases, one discretizes the noise so that the discretized noise is determined by a finite number of parameters

    - in the white noise case, the number of parameters has to increase as the spatial and/or temporal resolutions of the numerical scheme used to solve the PDEs increases

    - in the colored noise case, the number of parameters needed to approximate a correlated random field can, in practice, be chosen independently of the spatial/temporal resolutions

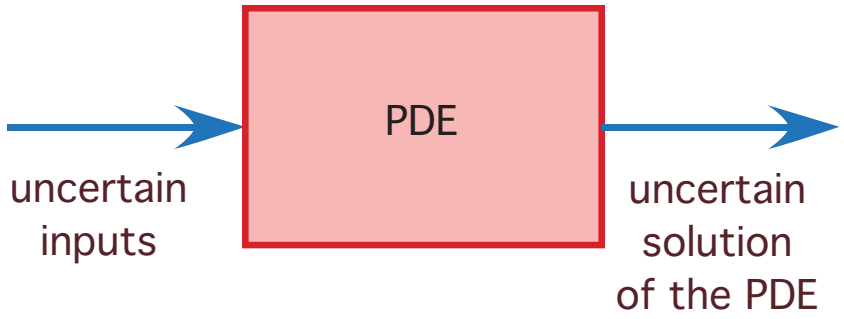
Uncertainty quantification

- Uncertainty quantification is the task of determining statistical information about outputs of a system, given statistical information about the inputs



  - of course, the system may have deterministic inputs as well

- We are interested in systems governed by partial differential equations

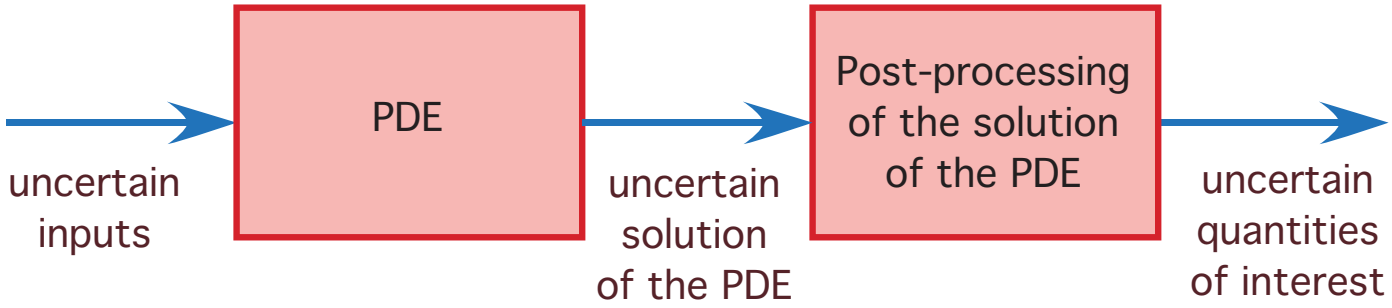


  - the solution of the partial differential equation defines the mapping from the input variables to the output variables

- Often, solutions of the PDE are not the primary output quantity of interest

  – quantities obtained by post-processing solutions of the PDE
    are more often of interest

      - of course, one still has to obtain a solution of
        the PDE to determine the quantity of interest



- A realization of the random system is determined by

  specifying a specific set of input variables

  and then

  using the PDE to determine the corresponding output variables

  – thus, a realization is a solution of a deterministic problem

- One is never interested in individual realizations of solutions of the PDE or of the quantities of interest

  – one is interested in determining statistical information about the quantities of interest, given statistical information about the inputs

Quantities of interest

- Suppose we have N random parameters $\{y_n\}_{n=1}^N$
  - we use the abbreviation $\vec{y} = \{y_1, y_2, \ldots, y_N\}$
  - each y_n could be distributed independently[†] according to its probability density function (PDF) $\rho_n(y_n)$ defined for y_n in a (possibly infinite) interval Γ_n
  - alternately, the parameters could be distributed according to a joint PDF $\rho(y_1, \ldots, y_N)$ that is a mapping from an N-dimensional set Γ into the real numbers
    - independently distributed parameters are the special case for which

$$\rho(y_1, \ldots, y_N) = \prod_{n=1}^{N} \rho_n(y_n) \qquad \text{and} \qquad \Gamma = \Gamma_1 \otimes \Gamma_2 \otimes \cdots \otimes \Gamma_N$$

[†]Without proper justification and sometimes incorrectly, it is almost always assumed that the parameters are independent; based on empirical evidence, sometimes this is a justifiable assumption in the parameters-are-"knobs" case, but for correlated random fields, it is justifiable only for the (spherical) Gaussian case; in general, independence is a simplifying assumption that is invoked for the sake of convenience, e.g., because of a lack of knowledge

- Realization = a solution $u(\mathbf{x}, t; \vec{y})$ of a PDE for a specific choice $\vec{y} = \{y_n\}_{n=1}^{N}$ for the random parameters
  - again, there is no interest in individual realizations

- One may be interested in statistics of solutions of the PDE
  - average or expected value

$$\overline{u}(\mathbf{x}, t) = \mathsf{E}[u(\mathbf{x}, t; \cdot)] = \int_{\Gamma} u(\mathbf{x}, t; \vec{y}) \rho(\vec{y})\, d\vec{y}$$

  - covariance

$$C_u(\mathbf{x}, t; \mathbf{x}', t') = \mathsf{E}\Big[\Big(u(\mathbf{x}, t; \cdot) - \overline{u}(\mathbf{x}, t)\Big)\Big(u(\mathbf{x}', t'; \cdot) - \overline{u}(\mathbf{x}', t')\Big)\Big]$$

$$= \int_{\Gamma} \Big(u(\mathbf{x}, t; \vec{y}) - \overline{u}(\mathbf{x}, t)\Big)\Big(u(\mathbf{x}', t'; \vec{y}) - \overline{u}(\mathbf{x}', t')\Big) \rho(\vec{y})\, d\vec{y}$$

  - variance $\quad C_u(\mathbf{x}, t; \mathbf{x}, t)$
  - higher moments

- One may instead be interested in statistics of spatial/temporal integrals of the solution of the PDE
  - for any fixed $\vec{y}$, we have, e.g.,

$$\mathcal{J}(t;\vec{y}) = \int_{\mathcal{D}} F(u;\vec{y})\, d\mathbf{x} \qquad \text{or} \qquad \mathcal{J}(\mathbf{x};\vec{y}) = \int_{t_0}^{t_1} F(u;\vec{y})\, dt$$

    or

$$\mathcal{J}(\vec{y}) = \int_{t_0}^{t_1} \int_{\mathcal{D}} F(u;\vec{y})\, d\mathbf{x} dt$$

    where $F(\cdot;\cdot)$ is given, $\mathcal{D}$ is a spatial domain, and (t_0, t_1) is a time interval

  - quantities defined with respect to integrals over boundary segments also often occur in practice

– examples

- the space-time average of u

$$\mathcal{J}(\vec{y}) = \int_{t_0}^{t_1} \int_{\mathcal{D}} u(\mathbf{x}, t; \vec{y})\, d\mathbf{x} dt$$

- if $\mathbf{u}$ denotes a velocity field, then

$$\mathcal{J}(t; \vec{y}) = \int_{\mathcal{D}} \mathbf{u}(\mathbf{x}, t; \vec{y}) \cdot \mathbf{u}(\mathbf{x}, t; \vec{y})\, d\mathbf{x}$$

is proportional to the kinetic energy

– again, one is not interested in the values of these quantities for specific choices of the parameters $\vec{y}$

- one is interested in their statistics

– example: expected value of the kinetic energy

$$\mathsf{E}\left[\int_{\mathcal{D}} \mathbf{u}(\mathbf{x},t;\vec{y})\cdot\mathbf{u}(\mathbf{x},t;\vec{y})\,d\mathbf{x}\right]$$

$$= \int_{\Gamma}\int_{\mathcal{D}} \mathbf{u}(\mathbf{x},t;\vec{y})\cdot\mathbf{u}(\mathbf{x},t;\vec{y})\rho(\vec{y})\,d\mathbf{x}\,d\vec{y}$$

- Thus, quantities of interest of this common type involve integrals over the parameter space[†]

  – e.g., for some $G(\cdot)$, integrals of the type

$$\int_{\Gamma} G\big(u(\mathbf{x},t;\vec{y})\big)\rho(\vec{y})\,d\vec{y} \qquad \text{or possibly} \qquad \int_{\Gamma} G\big(u(\mathbf{x},t;\vec{y});\mathbf{x},t,\vec{y}\big)\rho(\vec{y})\,d\vec{y}$$

[†]An important class of quantities of interest that arises in, e.g., reliability studies, but that we do not have time to consider involves integrals over a subset or Γ; in particular, we have

$$\int_{\Gamma} \chi_{u_0} G\big(u(\mathbf{x};\vec{y})\big)\rho(\vec{y})\,d\vec{y} = \int_{\Gamma_{u_0}} G\big(u(\mathbf{x};\vec{y})\big)\rho(\vec{y})\,d\vec{y}$$

where, for some given u_0

$$\chi_{u_0} = \begin{cases} 1 & \text{if } u(\mathbf{x};\vec{y}) \geq u_0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \Gamma_{u_0} = \{\vec{y}\in\Gamma \text{ such that } u(\mathbf{x};\vec{y}) \geq u_0\}$$

- Ideally, one wants to determine an approximation of the PDF for the quantity of interest,

  i.e., more than just a few statistical moments

  of some output quantity

  - the quantity of interest is a PDF

  - one way (but not the only way) to construct the approximate PDF is to compute many statistical moments of the output quantity

    - so, again, we are faced with evaluating stochastic integrals

Quadrature rules for stochastic integrals

- Integrals of the type

$$\int_{\Gamma} G\big(u(\mathbf{x},t;\vec{y})\big)\rho(\vec{y})\,d\vec{y}$$

cannot, in general, be evaluated exactly

- Thus, these integrals are approximated using a quadrature rule

$$\int_{\Gamma} G\big(u(\mathbf{x},t;\vec{y})\big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q G\big(u(\mathbf{x},t;\vec{y}_q)\big)\rho(\vec{y}_q)$$

for some choice of

quadrature weights $\{w_q\}_{q=1}^{Q}$ (real numbers)

and

quadrature points $\{\vec{y}_q\}_{q=1}^{Q}$ (points in the parameter domain Γ)

– Alternately, sometimes the probability density function is used in the determination of the quadrature points and weights so that instead one ends up with the approximation

$$\int_{\Gamma} G\big(u(\mathbf{x},t;\vec{y})\big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q G\big(u(\mathbf{x},t;\vec{y}_q)\big)$$

- Monte Carlo integration – the simplest rule $\Longrightarrow$
  - randomly select Q points in Γ according to the PDF $\rho(\vec{y})$
  - evaluate the integrand at each of the sample points
  - average the values so obtained
    - i.e., for all q, $w_q = 1/Q$
      - more on Monte Carlo and other quadrature rules later

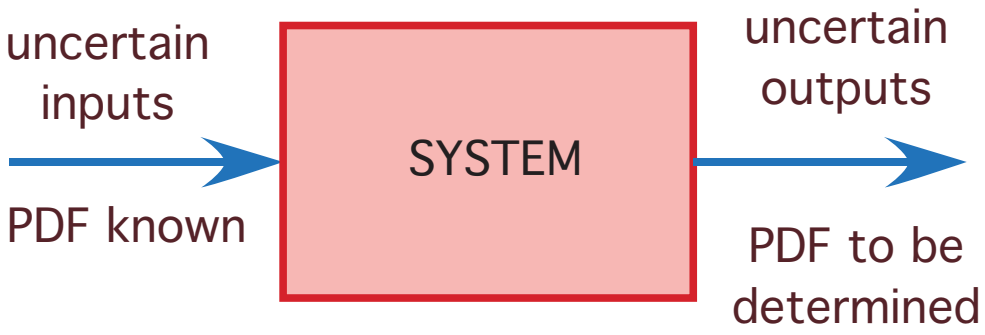
Big problem

- In practice, one usually does not know much about the statistics of the input variables
  - one is lucky if one knows a range of values, e.g., maximum and minimum values, for an input parameter
    - in which case one often assumes that the parameter is uniformly distributed over that range
  - if one is luckier, one knows the mean and variance for the input parameter
    - in which case one often assumes that the parameter is normally distributed
  - of course, one may be completely wrong in assuming such simple probability distributions for a parameter
- This leads to the need to solve stochastic model calibration problems

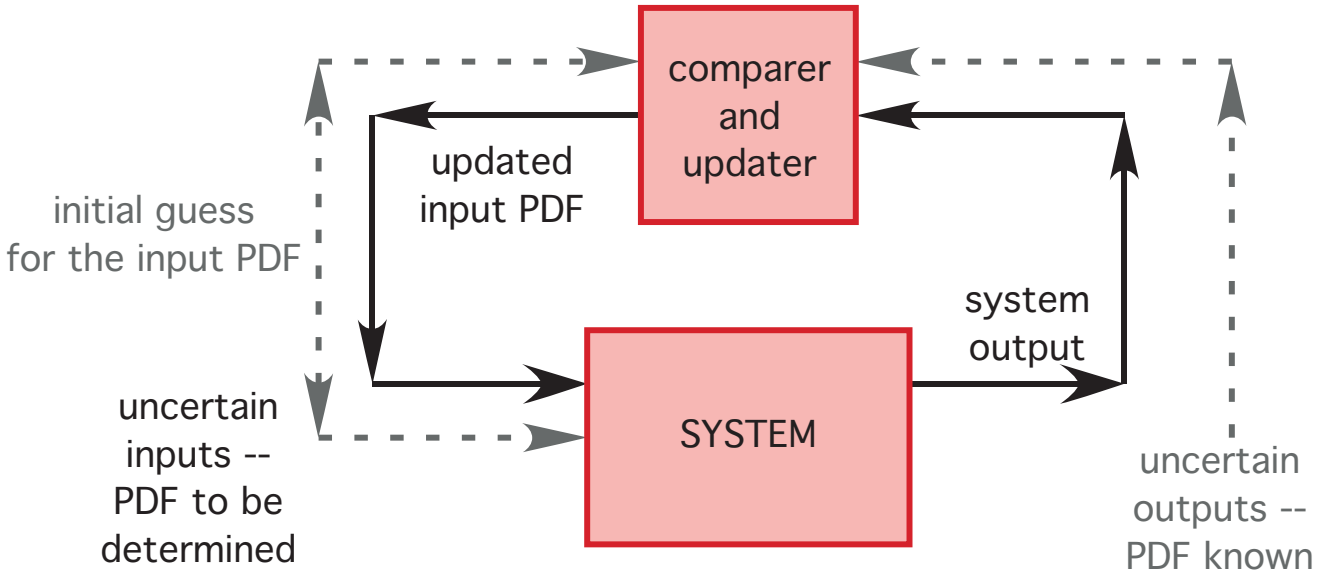
Model calibration

- Model calibration is the task of determining statistical information about the inputs of a system, given statistical information about the outputs
  - e.g., one can use experimental observations to determine the statistical information about the outputs
  - in particular, one wants to identify the probability density functions (PDF) of the input variables

- Of course, the system still maps the inputs to the outputs
  - thus, determining the input PDF is an inverse problem
  - usually involves an iteration in which guesses for the input PDF are updated
  - several ways to do the update, e.g., Baysean, maximum likelyhood, ...



Uncertainty quantification – direct problem

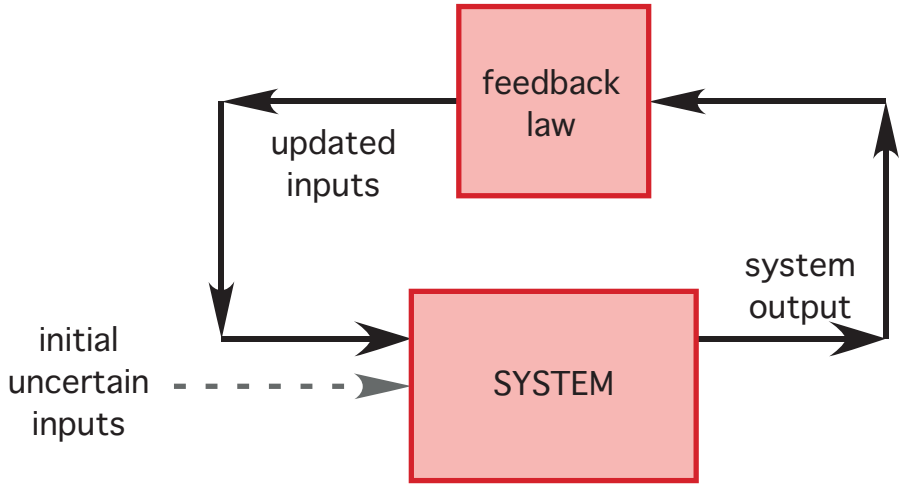


Model calibration – inverse problem

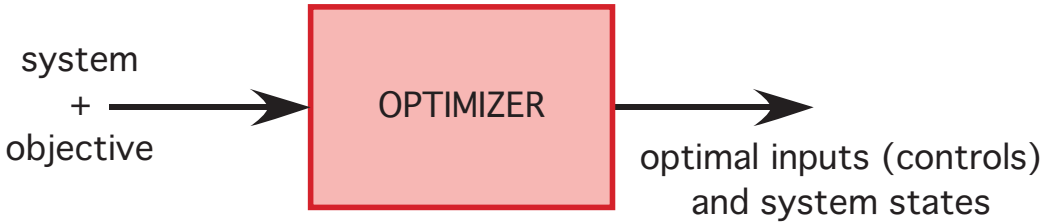
- Model calibration problems are a particular case of more general

  stochastic inverse, or parameter identification, or

  control, or optimization problems



Feedback control



Optimal control

OBSERVATIONS ABOUT THESE LECTURES

- Of greatest interest (to us) are nonlinear problems; however
  - so we focus on methods that are useful in the nonlinear setting
  - however, we do sometimes comment on special features of some methods that only hold for linear problems
- Both time-dependent and steady-state problems are of interest
  - for the sake of simplifying the exposition, we consider mostly steady-state problems
  - however, almost everthing we have to say applies equally well to time-dependent problems

WHITE NOISE

UNCORRELATED RANDOM FIELDS

- White noise refers to the case of uncorrelated random fields $\eta(\mathbf{x}, t; \omega)$ for which we have[†]

$$\mathsf{E}\big(\eta(\mathbf{x}, t; \omega)\big) = 0 \qquad \text{and} \qquad \mathsf{E}\big(\eta(\mathbf{x}, t; \omega)\eta(\mathbf{x}', t'; \omega)\big) = \delta(t - t')\delta(\mathbf{x} - \mathbf{x}')$$

  - at every point in space and at every instant in time, $\eta(\mathbf{x}, t; \omega)$ is independent and identically distributed

    - one determines $\eta(\mathbf{x}, t; \omega)$ at any point in space and any instant in time by sampling according to a given probability distribution

  - the Gaussian case is the one that often arises in practice (or because of a lack of information)

[†]The zero mean and unit variance assumptions are not restrictive

Discretizing white noise

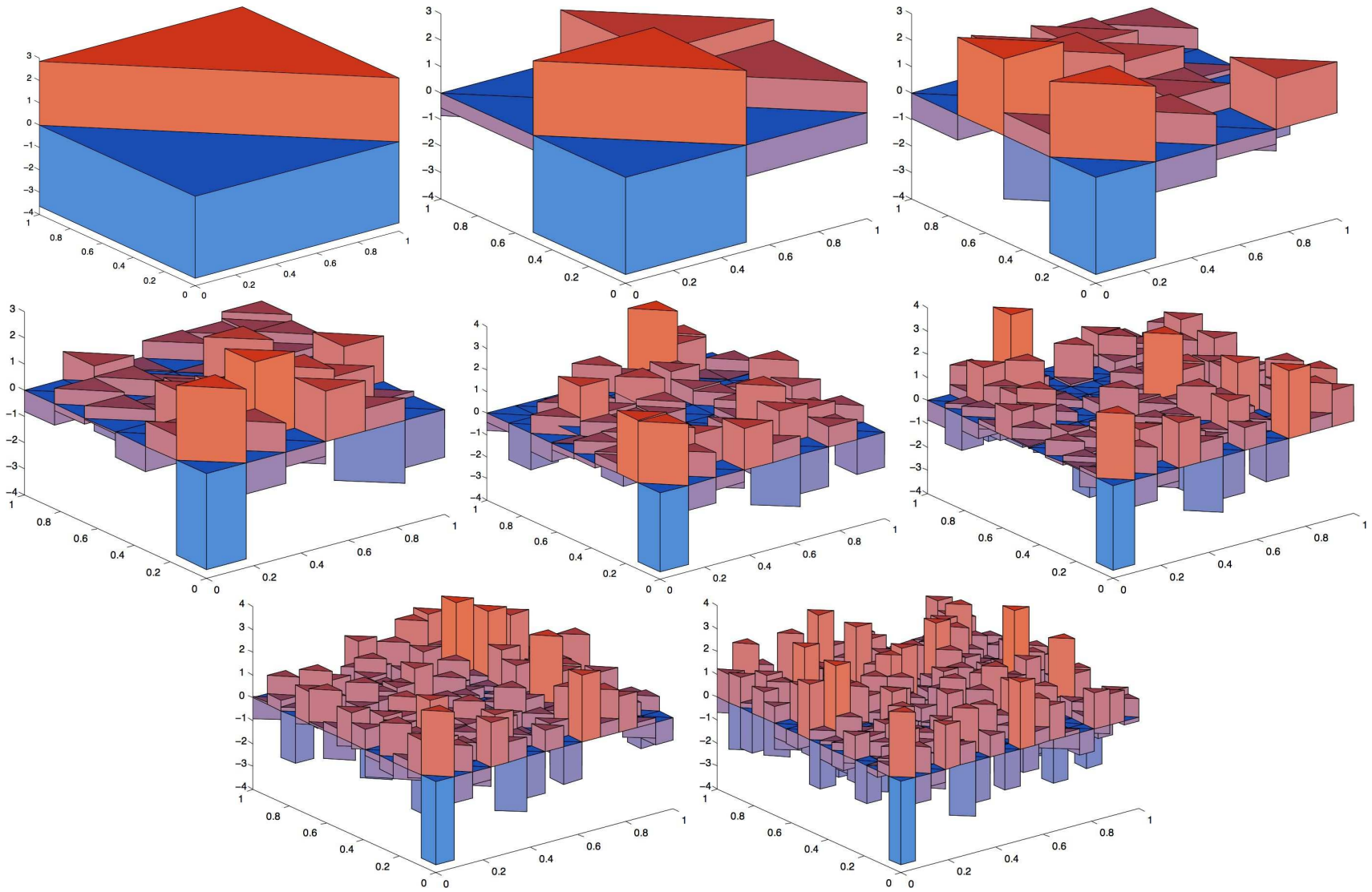
- In computer simulations, one cannot sample the Gaussian distribution at every point of the spatial domain and at every instant of time
  - white noise terms are replaced by discretized white noise terms
    - discretized white noise is more regular that white noise
- Among the means available for discretizing white noise, grid-based methods are the most popular

- To define a single realization of the discretized white noise, we
  - subdivide the spatial domain $\mathcal{D}$ into N_{space} subdomains
  - subdivide the temporal interval $[0, T]$ into N_{time} time subintervals
  - then, in the n_s-th spatial subdomain having volume V_{n_s} and in the n_t-th temporal subinterval having duration Δt_{n_t}, set

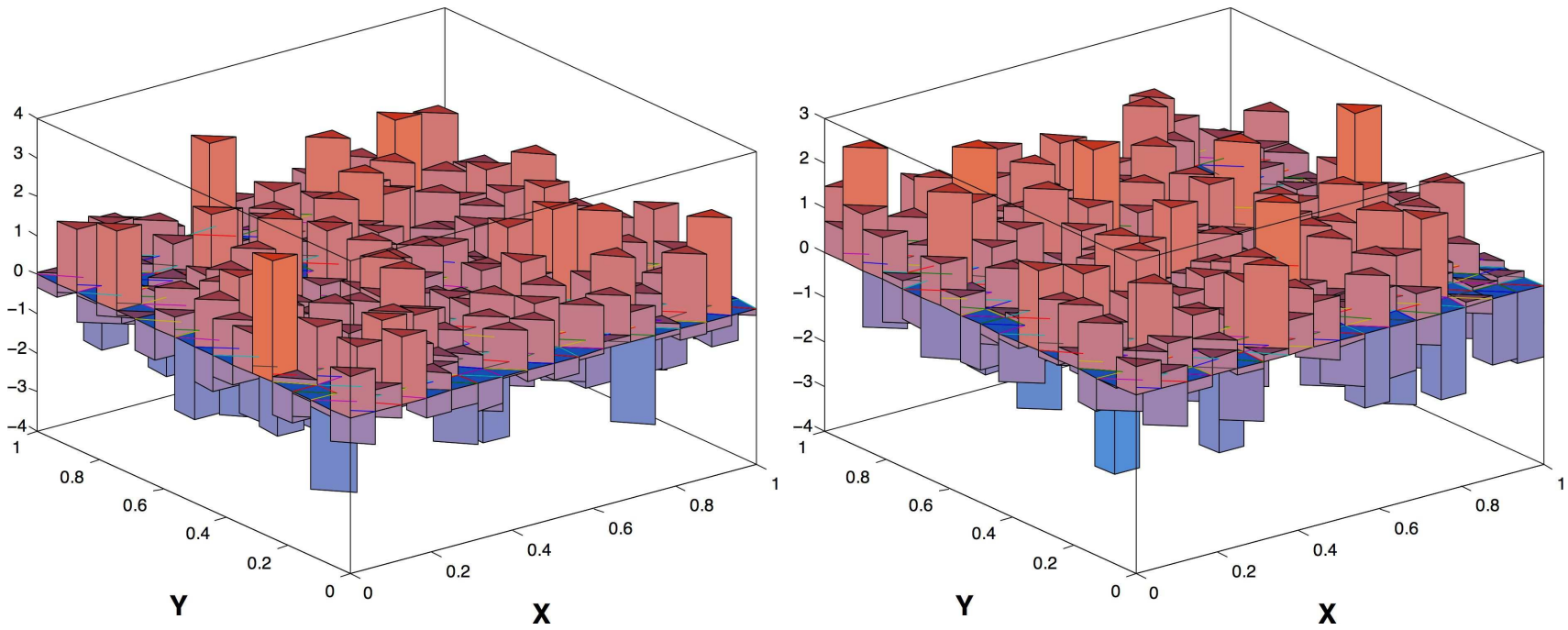
$$\eta_{approximate}(\mathbf{x}, t; \{y_{n_s,n_t}\}) = \frac{1}{\sqrt{\Delta t_{n_t}}\sqrt{V_{n_s}}} y_{n_s,n_t}$$

    where y_{n_s,n_t} are independent Gaussian samples having zero mean and unit variance

- Additional realizations are defined by resampling over the space-time grid



Realizations of discretized white noise at a same time interval in a square subdivided into 2, 8, 32, 72, 238, 242, 338, and 512 triangles



Realizations of discretized white noise at two different time intervals in a square subdivided into the same number of triangles

- Thus, the discretized white noise is piecewise constant in space and time

- Note that the piecewise constant function is much smoother than the random field it approximates

- It can be shown that

$$\lim_{N_{space}\to\infty,\, N_{time}\to\infty} \mathsf{E}\Big(\eta_{approximate}(\mathbf{x},t;\{y_{n_s,n_t}\})\,\eta_{approximate}(\mathbf{x}',t';\{y_{n_s,n_t}\})\Big)$$

$$= \mathsf{E}\big(\eta(\mathbf{x},t)\eta(\mathbf{x}',t')\big) = \delta(\mathbf{x}-\mathbf{x}')\delta(t-t')$$

- The white noise case has been reduced to a case of a large but finite number of parameters

  – we have the

  $N = N_{space}N_{time}$ parameters y_{n_s,n_t}

  where $n_s = 1,\ldots,N_{space}$ and $n_t = 1,\ldots,N_{time}$

  – if we refine the spatial grid and/or reduce the time step, the number of parameters increases

PDE'S FORCED BY WHITE NOISE

- Formally, we can write an evolution equation with white noise forcing as

$$\frac{\partial u}{\partial t} = A(u; \mathbf{x}, t) + f(\mathbf{x}, t) + B(u; \mathbf{x}, t)\eta(\mathbf{x}, t; \omega) \qquad \text{in } \mathcal{D} \times (0, T]$$

where

A is a possibly nonlinear deterministic operator

f is a deterministic forcing function

B is a possibly nonlinear deterministic operator

η is the white noise forcing function

– among many other cases,

A, f, and B can take care of cases with means $\neq 0$ and variances $\neq 1$

- If B is independent of u, we have additive white noise

$$\frac{\partial u}{\partial t} = A(u; \mathbf{x}, t) + f(\mathbf{x}, t) + b(\mathbf{x}, t)\, \eta(\mathbf{x}, t)$$

  - in practice, often b is a constant

- If B depends on u, we have multiplicative white noise

  - of particular interest is the case of B linear in u

$$\frac{\partial u}{\partial t} = A(u; \mathbf{x}, t) + f(\mathbf{x}, t) + b(\mathbf{x}, t) u\, \eta(\mathbf{x}, t)$$

- Some observations
  - solutions are not sufficiently regular for the equations just written to make sense
    - the renowned Ito calculus is introduced to make sense of differential equations with white noise forcing
  - white noise need not be restricted to forcing terms in the PDE
    - in practice, it can also appear
      in the coefficients of the PDEs and boundary and initial conditions
      in the data in boundary and initial conditions
      in the definition of the domain

- Spatial discretization of the PDE can be effected via a finite element method based on a triangulation of the spatial domain $\mathcal{D}$; temporal discretization is effected via a finite difference method, e.g., a backward Euler method

  – it is natural to use the same grids in space and time as are used to discretize the white noise

  – thus, if one refines the finite element grid and the time step, one also refines the grid and time step for the white noise discretization

- Once a realization of the discretized noise is chosen,

  i.e., once one chooses the $N_{space}N_{time}$ Gaussian samples η_{n_s,n_t},

  a realization of the solution of the PDE is determined

  by solving a deterministic problem

- For example, consider the problem

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta u + f(\mathbf{x},t) + b(\mathbf{x},t)u\,\eta(\mathbf{x},t;\omega) & \text{in } \mathcal{D}\times(0,T] \\ u = 0 & \text{in } \partial\mathcal{D}\times(0,T] \\ u(\mathbf{x},0) = u_0(\mathbf{x}) & \text{in } \mathcal{D} \end{cases}$$

  - subdivide $[0,T]$ into N_{time} subintervals of duration Δt_{n_t}, $n_t = 1,\ldots,N_{time}$

  - subdivide $\mathcal{D}$ into N_{space} finite elements $\{\mathcal{D}_{n_s}\}_{n_s=1}^{N_{space}}$

  - define a finite element space $S_0^h \subset H_0^1(\mathcal{D})$ with respect to the grid $\{\mathcal{D}_{n_s}\}_{n_s=1}^{N_{space}}$

– choose an approximation $u_{(0,h)}(\mathbf{x})$ to the initial data $u_0(\mathbf{x})$

– sample, from a standard Gaussian distribution,
the $N_{space}N_{time}$ values y_{n_s,n_t}, $n_s = 1, \ldots, N_{space}$ and $n_t = 1, \ldots, N_{time}$

– set $u_h^{(0)}(\mathbf{x}) = u_{(0,h)}(\mathbf{x})$

– then, for $n_t = 1, \ldots, N_{time}$, determine $u_h^{(n_t)}(\mathbf{x}) \in S_0^h$ from

$$\int_{\mathcal{D}} \frac{u_h^{(n_t)} - u_h^{(n_t-1)}}{\Delta t_{n_t}} v_h \, d\mathbf{x} + \int_{\mathcal{D}} \nabla u_h^{(n)} \cdot \nabla v_h \, d\mathbf{x}$$

$$= \int_{\mathcal{D}} f v_h \, d\mathbf{x} + \frac{1}{\sqrt{\Delta t_{n_t}}\sqrt{A_{n_s}}} \sum_{n_s=1}^{N_s} \int_{\mathcal{D}_{n_s}} y_{n_s,n_t} v_h \, d\mathbf{x} \qquad \text{for all } v^h \in S_0^h$$

- note that we have used a backward-Euler time stepping scheme

- This is a standard discrete finite element system for the heat equation, albeit with an unusual right-hand side

- Due to the lack of regularity of solutions of PDE's with white noise,

  the usual notions of convergence

  of the approximate solution to the exact solution

  do not hold,

  even in expectation

  – one has to be satisfied with very weak notions of convergence

COLORED NOISE

CORRELATED RANDOM FIELDS

- We now consider correlated random fields $\eta(\mathbf{x}, t; \omega)$
  - at each point $\mathbf{x}$ in a spatial domain $\overline{\mathcal{D}}$ and at each instant t in an time interval $[t_0, t_1]$, the value of η is determined by a random variable ω whose values are drawn from a given probability distribution
  - however, unlike the white noise case, the covariance function of the random field $\eta(\mathbf{x}, t; \omega)$ does not reduce to delta functions
- In rare cases, a formula for the random field is "known"
  - again, we cannot sample the random field at every spatial and temporal point
  - on the other hand, unlike the white noise case, the fact that the random field is correlated implies that one can find a discrete approximation to the random field for which the number of degrees of freedom can be thought of as fixed, i.e., independent of the spatial and temporal grid sizes

- More often, only the

  mean[†] $\mu_\eta(\mathbf{x}, t)$

  and

  covariance function $\text{cov}_\eta(\mathbf{x}, t; \mathbf{x}', t')$

  are known for points $\mathbf{x}$ and $\mathbf{x}'$ in $\overline{\mathcal{D}}$ and time instants t and t' in $[t_0, t_1]$

  – in this case, what we do not have is a formula for $\eta(\mathbf{x}, t; \omega)$

  – thus, we cannot evaluate $\eta(\mathbf{x}, t; \omega)$ when we need to

  – for example, if $\eta(\mathbf{x}, t; \omega)$ is a coefficient or a forcing function in a PDE, then to determine an approximate realization of the PDE we need to evaluate $\eta(\mathbf{x}, t; \omega)$ for a specific choice of ω and at specific points $\mathbf{x}$ and specific instants of time t used in the discretized PDE

[†] We have that

$$\mu_\eta(\mathbf{x}, t) = \mathsf{E}\big((\eta(\mathbf{x}, t; \cdot)\big)$$

and

$$\text{cov}_\eta(\mathbf{x}, t; \mathbf{x}', t') = \mathsf{E}\Big(\big(\eta(\mathbf{x}, t; \cdot) - \mu_\eta(\mathbf{x}, t)\big)\big(\eta(\mathbf{x}', t'; \cdot) - \mu_\eta(\mathbf{x}', t')\big)\Big)$$

- Examples of covariance functions

$$\text{cov}(\mathbf{x}, t; \mathbf{x}', t') = e^{-|\mathbf{x}-\mathbf{x}'|/L - |t-t'|/T}$$

and

$$\text{cov}(\mathbf{x}, t; \mathbf{x}', t') = e^{-|\mathbf{x}-\mathbf{x}'|^2/L^2 - |t-t'|^2/T^2}$$

where L is the correlation length and T is the correlation time

- large $L, T \Longrightarrow$ long-range order
- small $L, T \Longrightarrow$ short-range order

- Note that covariance functions are symmetric and positive

- So, we have two cases

  – the more common case for which only the mean and covariance function of the random field are known

    - we would like to find a simple formula depending on only a few parameters whose mean and covariance function are approximately the same as the given mean and covariance function

  – the rare case for which the random field is given as a formula but we want to approximate it

    - we would like to approximate it using few random parameters, certainly with a number of parameters that is independent of the spatial and temporal grid sizes

    - of course, this case can be turned into the first case by determining the mean and covariance function of the given random field (this may or may not be a good idea)

- Among the known ways for doing these tasks, we will focus on perhaps the most popular $\Longrightarrow$

  the Karhunen-Loève (KL) expansion of a random field $\eta(\mathbf{x}, t; \omega)$

  – given the mean and covariance of a random field $\eta(\mathbf{x}, t; \omega)$,

  - the KL expansion provides a simple formula that
    can be used whenever one needs a value $\eta(\mathbf{x}, t; \omega)$

  – to keep things simple, we discuss KL expansions
  for the case of spatially-dependent random fields

The Karhunen-Loève expansion

- Given the mean $\mu_\eta(\mathbf{x})$ and covariance $\text{cov}_\eta(\mathbf{x}, \mathbf{x}')$ of a random field $\eta(\mathbf{x}; \omega)$, determine the eigenpairs $\{\lambda_n, b_n(\mathbf{x})\}_{n=1}^{\infty}$ from the eigenvalue problem

$$\int_{\mathcal{D}} \text{cov}_\eta(\mathbf{x}, \mathbf{x}')\, b(\mathbf{x}')\, d\mathbf{x}' = \lambda b(\mathbf{x})$$

  - often in practice, an approximate version of this problem is solved, e.g., using a finite element method

  - due to the symmetry of $\text{cov}_\eta(\cdot;\cdot)$, the eigenvalues λ_n are real and the eigenfunctions $b_n(\mathbf{x})$ can be chosen to be real and orthonormal, i.e.,

$$\int_{\mathcal{D}} b_n(\mathbf{x})\, b_{n'}(\mathbf{x})\, d\mathbf{x} = \delta_{nn'}$$

  - due to the positivity of $\eta(\mathbf{x}; \omega)$, the eigenvalues are all positive

    - without loss of generality, they may be ordered in non-increasing order $\lambda_1 \geq \lambda_2 \geq \cdots$

- Then, the random field $\eta(\mathbf{x};\omega)$ admits the KL expansion[†]

$$\eta(\mathbf{x};\omega) = \mu_\eta(\mathbf{x}) + \sum_{n=1}^{\infty} \sqrt{\lambda_n}\, b_n(\mathbf{x})\, Y_n(\omega)$$

where $\{Y_n(\omega)\}_{n=1}^{\infty}$ are centered and uncorrelated random variables, i.e.,

$$\mathsf{E}\big(Y_n(\omega)\big) = 0 \qquad \mathsf{E}\big(Y_n(\omega) Y_{n'}(\omega)\big) = 0$$

that inherit the probability structure of the random field $\eta(\mathbf{x};\omega)$

  - e.g., if $\eta(\mathbf{x};\omega)$ is a Gaussian random field, then the Y_n's are all Gaussian random variables

[†] To see this, let us make the ansatz

$$\eta(\mathbf{x};\omega) = \mu_\eta(\mathbf{x}) + \sum_{n=1}^{\infty} \alpha_n b_n(\mathbf{x}) y_n(\omega)$$

where

$$\int_{\mathcal{D}} b_n(\mathbf{x}) b_{n'}(\mathbf{x})\, d\mathbf{x} = \delta_{nn'}, \qquad \mathsf{E}(y_n) = 0, \qquad \text{and} \qquad \mathsf{E}(y_n y_{n'}) = \delta_{nn'}$$

i.e., $\{b_n(\cdot)\}_{n=1}^{\infty}$ is a set of orthonormal functions and $\{y_n(\cdot)\}_{n=1}^{\infty}$ is a set of uncorrelated random variables; we then have that

$$\mathsf{E}(\eta) = \mu_\eta(\mathbf{x}) + \sum_{n=1}^{\infty} \alpha_n b_n(\mathbf{x}) \mathsf{E}(y_n) = \mu_\eta(\mathbf{x})$$

and

$$\mathsf{E}\Big(\big(\eta(\mathbf{x};\cdot) - \mu_\eta(\mathbf{x})\big) \big(\eta(\mathbf{x}';\cdot) - \mu_\eta(\mathbf{x}')\big) \Big) = \sum_{n=1}^{\infty} \sum_{n'=1}^{\infty} \alpha_n \alpha_{n'} b_n(\mathbf{x}) b_{n'}(\mathbf{x}') \mathsf{E}(y_n y_{n'}) = \sum_{n=1}^{\infty} \alpha_n^2 b_n(\mathbf{x}) b_n(\mathbf{x}')$$

so that

$$\mathsf{cov}_\eta(\mathbf{x},\mathbf{x}') = \sum_{n=1}^{\infty} \alpha_n^2 b_n(\mathbf{x}) b_n(\mathbf{x}');$$

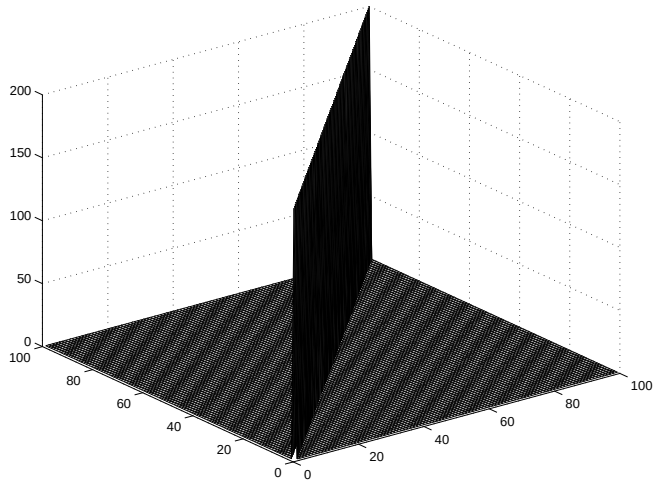
then, we have that

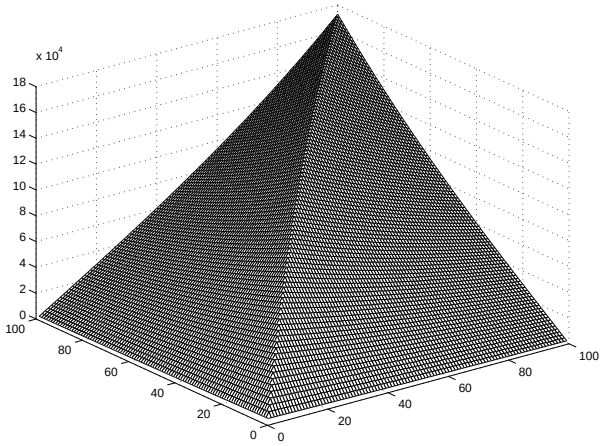
$$\int_{\mathcal{D}} \mathsf{cov}_\eta(\mathbf{x},\mathbf{x}') b_{n'}(\mathbf{x}')\, d\mathbf{x}' = \sum_{n=1}^{\infty} \alpha_n^2 b_n(\mathbf{x}) \int_{\mathcal{D}} b_n(\mathbf{x}') b_{n'}(\mathbf{x}')\, d\mathbf{x}' = \alpha_{n'}^2 b_{n'}(\mathbf{x})$$

so that indeed $\{\alpha_n^2, b_n(\mathbf{x})\}_{n=1}^{\infty}$ are the eigenpairs, i.e., we recover the KL expansion

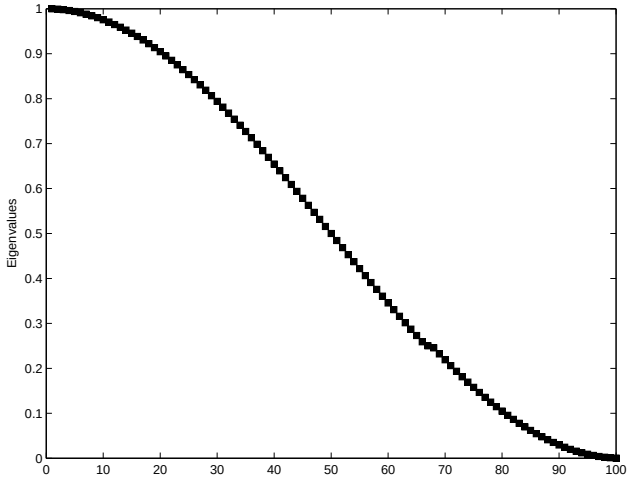
- The usefulness of the KL expansion results from the fact that the eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ decay as n increases

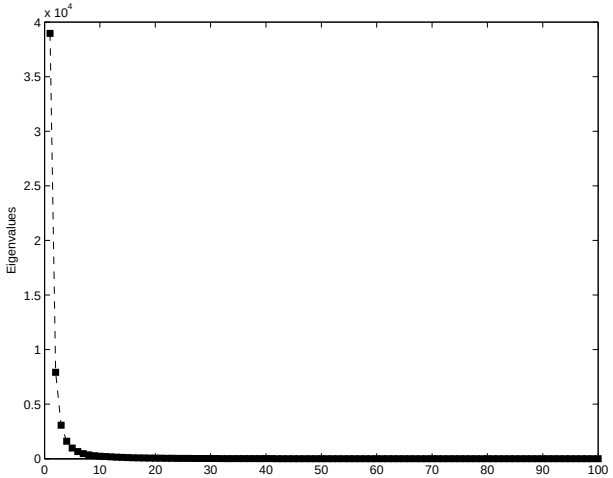
  – how fast they decay depends on the smoothness of the covariance function $\text{cov}_\eta(\mathbf{x}, \mathbf{x}')$ and on the correlation length L





Peaked and smooth covariance functions





Corresponding KL eigenvalues

- The decay of the eigenvalues implies that truncated KL expansions

$$\eta_N(\mathbf{x};\omega) = \mu(\mathbf{x}) + \sum_{n=1}^{N} \sqrt{\lambda_n} b_n(\mathbf{x}) Y_n(\omega)$$

can be accurate approximations to the exact expansions

– if one wishes for the relative error to be less than a prescribed tolerance δ, i.e., if one wants

$$\frac{\|\eta_N - \eta\|^2}{\|\eta\|^2} \leq \delta,$$

one should choose N to be the smallest integer such that

$$\frac{\sum_{n=N+1}^{\infty} \lambda_n}{\sum_{n=1}^{\infty} \lambda_n} \leq \delta \qquad \text{or, equivalently,} \qquad \frac{\sum_{n=1}^{N} \lambda_n}{\sum_{n=1}^{\infty} \lambda_n} \geq 1 - \delta$$

- Although the Y_n's are uncorrelated, in general they are not independent
  - in fact, they are independent if and only if they are (spherical) Gaussian
  - however, every random field can, in principle, be written as a function of a Gaussian random field
    - the inverse of the cumulative probability density of the given field,

    so that, in this way, we only have to deal with Gaussian random variables

- Dealing with independent random variables can have important practical consequences

- One important issue is the well posedness of the PDE when using a KL representation of random fields

  – suppose the coefficient $a(\mathbf{x};\omega)$ of an elliptic PDE is a random field

    - it cannot be a Gaussian random field since then it would admit negative values, which is not allowable

  – one way to get around this is to let, with $a_{min} > 0$,

$$a(\mathbf{x};\omega) = a_{min} + e^{\eta(\mathbf{x};\omega)}$$

  where $\eta(\mathbf{x};\omega)$ is a Gaussian random field with given mean and covariance

  – then, using a truncated KL expansion for $\eta(\mathbf{x};\omega)$, we have that

$$a(\mathbf{x};\omega) = a_{min} + e^{\mu(\mathbf{x}) + \sum_{n=1}^{N} \sqrt{\lambda_n}\, b_n(\mathbf{x})\, Y_n(\omega)}$$

  where $\{Y_n(\omega)\}_{n=1}^{N}$ are Gaussian random variables

Approximating Gaussian random fields

- For Gaussian random fields, we are done: we identify the random variables $\{Y_n(\omega)\}_{n=1}^N$ with Gaussian random parameters $\{y_n\}_{n=1}^N$ such that $\vec{y} \in \Gamma = \mathbb{R}^N$

- Let $\eta(\mathbf{x};\omega)$ be a Gaussian random field
  - we approximate $\eta(\mathbf{x};\omega)$ by its N-term truncated KL expansion

$$\eta_N(\mathbf{x};\omega) = \mu(\mathbf{x}) + \sum_{n=1}^{N} \sqrt{\lambda_n} b_n(\mathbf{x}) y_n$$

    where $\{y_n\}_{n=1}^N$ are Gaussian random parameters

- Thus, we now have a formula for the (approximation to a) random field that involves a finite number of random parameters
  - we can then use any of the methods to be discussed for problems involving a finite number of given random parameters to solve the problems described in terms of Gaussian random fields

Approximating non-Gaussian random fields

- If $\xi(\mathbf{x};\omega)$ is the given correlated random field and if the cumulative density function $F_\xi(\omega)$ is known, then one can write

$$\xi(\mathbf{x};\omega) = F_\xi^{-1}\big(\eta(\mathbf{x};\omega)\big) \qquad \text{where } \eta(\mathbf{x};\omega) \text{ is a Gaussian random field}$$

  - then, one can approximate $\eta(\mathbf{x};\omega)$ using a truncated KL expansion in terms of Gaussian random parameters $\{y_n\}_{i=1}^N$ so that

$$\xi_N(\mathbf{x};\omega) = F_\xi^{-1}\big((\eta_N(\mathbf{x};\omega)\big) = F_\xi^{-1}\Big(\mu(\mathbf{x}) + \sum_{n=1}^{N} \sqrt{\lambda_n} b_n(\mathbf{x}) y_n\Big)$$

  - so, again, we have obtained a formula for an approximation of the general random field $\xi(\mathbf{x};\omega)$ in terms of N random Gaussian parameters so that we can use any of the methods to be discussed for the random parameters case to find approximate solutions of the stochastic PDE

RANDOM PARAMETERS

PDE'S with random inputs depending on random parameters

- One or more

  input functions,

  e.g., coefficients, forcing terms, initial data, etc. in a

  partial differential equation

  depend on a finite number of random parameters

  - the input function could also depend on space and time
  - the random parameters could come from a Karhunen-Loève expansion of a correlated random field
  - the random parameters could appear naturally in the definition of input function

    - e.g., the Young's modulus or a diffusivity coefficient could be random

- Ideally, we would know the probability density function (PDF) for each parameter
  - as has already been mentioned, in practice, we know very little about the statistics of input parameters
  - however, we will assume that we know the PDFs for all the random input parameters

- Example: a nonlinear parabolic equation

$$c(\mathbf{x},t;y_{N_b},\ldots,y_{N_c})\frac{\partial u}{\partial t}-\nabla\cdot\Big(a(\mathbf{x},t;y_1,\ldots,y_{N_a})\nabla u\Big)+b(\mathbf{x},t;y_{N_a+1},\ldots,y_{N_b})u^3$$
$$=f(\mathbf{x},t;y_{N_c+1},\ldots,y_{N_f})\quad\text{on }\mathcal{D}(y_{N_i+1},\ldots,y_{N_g};y_{N_g+1},\ldots,y_{N_h})$$

$$u=f_{dir}(\mathbf{x},t;y_{N_f+1},\ldots,y_{N_d})\quad\text{on }\partial\mathcal{D}_D(y_{N_i+1},\ldots,y_{N_g})$$

$$a(\mathbf{x},t;y_1,\ldots,y_{N_a})\frac{\partial u}{\partial n}=f_{neu}(\mathbf{x},t;y_{N_d+1},\ldots,y_{N_e})\quad\text{on }\partial\mathcal{D}_N(y_{N_g+1},\ldots,y_{N_h})$$

$$u=f_0(\mathbf{x};y_{N_e+1},\ldots,y_{N_i})\quad\text{on }\mathcal{D}(y_{N_i+1},\ldots,y_{N_g};y_{N_g+1},\ldots,y_N)$$

  - the y_n's are random parameters
  - a, b, c, f, f_{dir}, f_{neu}, and f_0 are given functions of $\mathbf{x}$, t, and the random parameters
  - the boundary segments $\partial\mathcal{D}_D$ and $\partial\mathcal{D}_N$ are parametrized by the corresponding random parameters
  - of course, ∇ and $\nabla\cdot$ are operators involving spatial derivatives

- Concrete example: an elliptic PDE for $u(\mathbf{x}; y_1, \ldots, y_5)$

  – consider

$$\nabla \cdot \Big(a(\mathbf{x}; y_1, y_2) \nabla u \Big) = f(\mathbf{x}; y_3, y_4) \quad \text{on } \mathcal{D}(y_5)$$

$$u = 0 \quad \text{on } \partial\mathcal{D}(y_5)$$

where

$$a(\mathbf{x}; y_1, y_2) = 3 + |\mathbf{x}| \big(y_1^2 + \sin(y_2) \big)$$

$$f(\mathbf{x}; y_3, y_4) = y_3 e^{-y_4 |\mathbf{x}|^2}$$

$$\mathcal{D}(y_5) = (0, 1) \times (0, 1 + 0.3 y_5)$$

with

$$\rho_1(y_1) = N(0; 1) \qquad \rho_2(y_2) = U(0; 0.5\pi) \qquad \rho_3(y_3) = N(0; 2)$$

$$\rho_4(y_4) = U(0, 1) \qquad \rho_5(y_5) = U(-1, 1)$$

- The well-posedness of the PDE for all possible values of the parameters is a very important (and sometimes ignored) consideration

  – for the simple elliptic PDE

$$\nabla \cdot \Big(a(\mathbf{x}; y_1, \ldots, y_N) \nabla u \Big) = f(\mathbf{x}) \quad \text{on } \mathcal{D}$$

  we must have, for some $a_{max} \geq a_{min} > 0$,

$$a_{min} \leq a(\mathbf{x}; y_1, \ldots, y_{\widetilde{N}}) \leq a_{max} \qquad \text{for all } \mathbf{x} \in \mathcal{D} \text{ and all } \vec{y} \in \Gamma$$

  – this could place a constraint on how one chooses the PDF for the parameters

  – for example, if we have

$$a(\mathbf{x}; y) = a_0 + y$$

  where $a_0 > 0$, we cannot choose y to be a Gaussian random parameter

A brief taxonomy of methods for stochastic PDEs with random input parameters

- Stochastic finite element methods (SFEMs)

  $\Longrightarrow$ methods for which spatial discretization is effected using finite element methods[†]

- One particular class of SFEMs is known as stochastic Galerkin methods (SGMs)

  $\Longrightarrow$ methods for which probabilistic discretization is also effected using a Galerkin method

  – polynomial chaos and generalized polynomial chaos methods are SGMs

  – we will also consider other SGMs

[†] Throughout, we assume that spatial discretization is effected using finite element methods; most of what we say also holds for other spatial discretization approaches, e.g., finite differences, finite volumes, spectral, etc.

- Another class of SFEMs are stochastic sampling methods (SSMs)

  $\Longrightarrow$ points in the parameter domain Γ are sampled,
  then used as inputs for the PDE, and then
  ensemble averages of output quantities of interest are computed

  – Monte-Carlo finite element methods are the simplest SSMs

  – stochastic collocation methods (SCMs) are also SSMs

  - the sampling points are the quadrature points corresponding to some quadrature rule

Example used to describe numerical methods for SPDEs

- Let $\mathcal{D} \subset \mathbb{R}^d$ denote a spatial domain[†] with boundary $\partial\mathcal{D}$
  - $d = 1, 2$, or 3 denotes the spatial dimension
  - $\mathbf{x} \in \mathcal{D}$ denotes the spatial variable

- Let $\Gamma \in \mathbb{R}^N$ denote a parameter domain
  - N denotes the number of parameters
  - $\vec{y} = (y_1, y_2, \ldots, y_N) \in \Gamma$ denotes the random parameter vector
  - note that we have a finite number of parameters $\{y_n\}_{n=1}^N$
    but they can take on values anywhere in the Euclidean domain Γ

[†]For the sake of simplicity, we now consider stationary problems; all we have to say holds equally well for time-dependent problems

- Let $u(\mathbf{x};\vec{y}) \in X \times Z$ denote the solution of the SPDE[†‡]
  - generally, $Z = L^q_\rho(\Gamma)$, the space of functions of N variables whose q-th power is integrable with respect to the joint PDF (the weight function) $\rho(\cdot)$, i.e., those functions $g(\vec{y})$ for which

$$\int_\Gamma |g(\vec{y})|^q \rho(\vec{y})\, d\vec{y} < \infty$$

    - q is chosen according to how many statistical moments one wants to have well defined
    - the most common choice is $q = 2$ so that up to the second moments are well defined
    - if $\{y_1, \ldots, y_N\}$ are independent and if $L^q_{\rho_n}(\Gamma_n)$ denotes the space of functions that have integrable q-th powers with respect to the PDF $\rho_n(y_n)$, we have that

$$L^q_\rho(\Gamma) = L^q_{\rho_1}(\Gamma_1) \otimes L^q_{\rho_2}(\Gamma_2) \otimes \cdots \otimes L^q_{\rho_N}(\Gamma_N)$$

[†]Often, X is a Sobolev space such as $H^1_0(\mathcal{D})$

[‡]It is not always convenient to use a product space $X \times Z$; for example, it may make more sense to have $u \in L^q_\rho(\Gamma; X)$

- It is entirely natural to then treat a function $u(\mathbf{x};\vec{y})$ of d spatial variables and of N random parameters as a function of $d+N$ variables

- This leads one to consider a Galerkin weak formulation in physical and parameter space: seek $u(\mathbf{x};\vec{y}) \in X \times Z$

$$\int_\Gamma \int_\mathcal{D} S(u;\vec{y})T(v)\rho(\vec{y})\,d\mathbf{x}d\vec{y} = \int_\Gamma \int_\mathcal{D} vf(\vec{y})\rho(\vec{y})\,d\mathbf{x}d\vec{y} \qquad \forall\, v \in X \times Z$$

where[†]

  - $S(\cdot;\cdot)$ is, in general, a nonlinear operator[‡]
  - $T(\cdot)$ is a linear operator

[†]Of course, if $\mathrm{E}(\cdot)$ denotes the expected value, this may be expressed in the form

$$\mathrm{E}\left(\int_\mathcal{D} S(u;\vec{y})T(v)\rho(\vec{y})\,d\mathbf{x} - \int_\mathcal{D} vf(\vec{y})\rho(\vec{y})\,d\mathbf{x}\right) = 0$$

[‡]S, T, and f could also depend on $\mathbf{x}$, but we do not explicitly keep track of such dependences

- In general, we would have a sum of such terms, i.e., we would have that

$$\sum_{m=1}^{M} \int_{\Gamma} \int_{\mathcal{D}} S_m(u;\vec{y}) T_m(v) \rho(\vec{y})\, d\mathbf{x} d\vec{y}$$

$$= \int_{\Gamma} \int_{\mathcal{D}} v f(\vec{y}) \rho(\vec{y})\, d\mathbf{x} d\vec{y} \qquad \forall\, v \in X \times Z$$

  - however, without loss of generality, it suffices for our purposes to consider the simpler single-term form

$$\int_{\Gamma} \int_{\mathcal{D}} S(u;\vec{y}) T(v) \rho(\vec{y})\, d\mathbf{x} d\vec{y} = \int_{\Gamma} \int_{\mathcal{D}} v f(\vec{y}) \rho(\vec{y})\, d\mathbf{x} d\vec{y} \qquad \forall\, v \in X \times Z$$

- In general,
  - both S and T could involve derivatives with respect to $\mathbf{x}$
  - but S does not involve derivatives with respect to $\vec{y}$

- Example

  – suppose our SPDE problem is given by

$$-\nabla \cdot \Big(a(\vec{y})\nabla u\Big) + c(\vec{y})u^3 = f(\vec{y}) \qquad \text{in } \mathcal{D} \qquad \text{and} \qquad u = 0 \qquad \text{in } \partial\mathcal{D}$$

    - of course, a, c, and f could also depend on $\mathbf{x}$

  – we then have that $X = H_0^1(\mathcal{D})$ and $Z = L_\rho^2(\Gamma)$ and the weak formulation:

    - seek $u(\mathbf{x}; \vec{y}) \in H_0^1(\mathcal{D}) \times L_\rho^2(\Gamma)$ such that

$$\int_{\mathcal{D}} \int_{\Gamma} \Big(a(\vec{y})\nabla u\Big) \cdot \nabla v \rho(\vec{y})\, d\vec{y} d\mathbf{x} + \int_{\mathcal{D}} \int_{\Gamma} \Big(c(\vec{y})u^3\Big) v \rho(\vec{y})\, d\vec{y} d\mathbf{x}$$
$$= \int_{\mathcal{D}} \int_{\Gamma} f(\vec{y}) v \rho(\vec{y})\, d\vec{y} d\mathbf{x} \qquad \forall\, v \in H_0^1(\mathcal{D}) \times L_\rho^2(\Gamma)$$

  – in the first term, we have that $S(u, \vec{y}) = a(\vec{y})\nabla u$ and $T = \nabla v$

  – in the second term, we have that $S(u, \vec{y}) = c(\vec{y})u^3$ and $T = v$

- We assume that all methods considered use the same approach to effect discretization with respect to the spatial variables

  – we focus on finite element methods,

    i.e., on stochastic finite element methods

  – throughout, $\{\phi_j(\mathbf{x})\}_{j=1}^{J}$ denotes a basis for the finite element space $X_J \subset X$ used to effect spatial discretization

    - note that J denotes the dimension of the finite element space

- We assume that Γ is a parameter box

  - without loss of generality, it can be taken to be a hypercube in $\mathbb{R}^N$

  - for parameters with unbounded PDFs, Γ can be of infinite extent

  - if the parameters are constrained, Γ need not be so simple

    e.g., if y_1 and y_2 are independent except that we require that $y_1^2 + y_2^2 \leq 1$, then Γ would be the unit circle

STOCHASTIC GALERKIN METHODS

STOCHASTIC GALERKIN METHODS

- Functions of the parameters have to be discretized in much the same way functions of the (finite number of) spatial variables have to be discretized
  - spatial discretization is effected via a standard finite element discretization in the usual manner by choosing a J-dimensional subspace $X_J \subset X$
  - let $\{\phi_j(\vec{y})\}_{j=1}^{J}$ denote a basis for X_J
- Stochastic Galerkin methods are methods for which discretization with respect to parameter space is also effected using a Galerkin approach, i.e.,
  - we choose a K-dimensional subspace $Z_K \subset Z$
  - let $\{\psi_k(\vec{y})\}_{k=1}^{K}$ denote a basis for the parameter approximating space Z_K

- Due to the product nature of the domain $\mathcal{D} \otimes \Gamma$ and of the space $X \otimes Z$, it is natural to seek approximations that use this structure, i.e.,
  - approximations are defined as a sum of products of the spatial and probabilistic basis functions

- Thus, we seek an approximate solution of the SPDE of the form[†]

$$u_{JK} = \sum_{j=1}^{J} \sum_{k=1}^{K} c_{jk} \phi_j(\mathbf{x}) \psi_k(\vec{y}) \in X_J \times Z_K$$

- The coefficients c_{jk}, and therefore u_{JK}, are determined by solving the problem

$$\int_{\mathcal{D}} \int_{\Gamma} \rho(\vec{y}) S(u_{JK}, \vec{y}) T(v) \, d\mathbf{x} d\vec{y} = \int_{\mathcal{D}} \int_{\Gamma} \rho(\vec{y}) v f(\vec{y}) \, d\mathbf{x} d\vec{y} \qquad \forall \, v \in X_J \times Z_K$$

[†]Potentially, some economies can be effected if one also approximates the data functions (e.g., coefficients) appearing in the problem in the same way one approximates the solution, e.g., for a data function $a(\mathbf{x}; \vec{y})$, one determines $a_k(\mathbf{x})$, $k = 1, \ldots, K$, such that

$$\sum_{k=1}^{K} a_k(\mathbf{x}) \psi_k(\vec{y}) \approx a(\mathbf{x}; y_1, \ldots, y_N);$$

in actuality, these economies can be realized only in very limited settings; more on this later

- We then have that the discretized problem

$$\int_{\mathcal{D}}\int_{\Gamma}\rho(\vec{y})S\Big(\sum_{j=1}^{J}\sum_{k=1}^{K}c_{jk}\phi_j(\mathbf{x})\psi_k(\vec{y}),\vec{y}\Big)T\Big(\phi_{j'}(\mathbf{x})\Big)\psi_{k'}(\vec{y})\,d\mathbf{x}d\vec{y}$$
$$=\int_{\mathcal{D}}\int_{\Gamma}\rho(\vec{y})\phi_{j'}(\mathbf{x})\psi_{k'}(\vec{y})f(\vec{y})\,d\mathbf{x}d\vec{y}$$

for $j' \in \{1,\ldots,J\}$ and $k' \in \{1,\ldots,K\}$

- Of course, the solution

$$u_{JK}(\mathbf{x};\vec{y}) = \sum_{j=1}^{J}\sum_{k=1}^{K}c_{jk}\phi_j(\mathbf{x})\psi_k(\vec{y})$$

of this problem is independent of the basis set used

– although the coefficients c_{jk} do depend on the choice of basis

- In general, the integrals cannot be evaluated exactly
  - quadrature rules must be invoked to effect approximate evaluations
  - thus, the integrals with respect to the parameter domain[†] Γ are approximated by a quadrature rule to obtain

$$\sum_{r=1}^{R} \widehat{w}_r \rho(\widehat{\vec{y}}_r) \psi_{k'}(\widehat{\vec{y}}_r) \int_{\mathcal{D}} S\Big(\sum_{j=1}^{J} \sum_{k=1}^{K} c_{jk} \phi_j(\mathbf{x}) \psi_k(\widehat{\vec{y}}_r), \widehat{\vec{y}}_r \Big) T\Big(\phi_{j'}(\mathbf{x}) \Big) \, d\mathbf{x}$$

$$= \sum_{r=1}^{R} \widehat{w}_r \rho(\widehat{\vec{y}}_r) \psi_{k'}(\widehat{\vec{y}}_r) \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\widehat{\vec{y}}_r) \, d\mathbf{x}$$

for $j' \in \{1, \ldots, J\}$ and $k' \in \{1, \ldots, K\}$

for some choice of quadrature weights $\{\widehat{w}_r\}_{r=1}^{R}$ and quadrature points $\{\widehat{\vec{y}}_r\}_{r=1}^{R}$ in Γ

[†]Integrals with respect to the spatial domain $\mathcal{D}$ must also be approximated using quadrature rules; we do not need to consider this issue since we assume that all methods discussed treat all aspects of the spatial discretization in the same manner

– this quadrature rule need not be the same as the quadrature rule $\{w_q, \vec{y}_q\}_{r=1}^{Q}$ used to obtain the approximation of a quantity of interest

- In general, the discrete problem is a fully coupled (in physical and parameter spaces) $JK \times JK$ system

  – there are JK equations and JK degrees of freedom c_{jk}[†]

- On the other hand, one can solve for the approximate dependence of the solution $u_{JK}(\mathbf{x}, \vec{y})$ on both the spatial coordinates $\mathbf{x}$ and the random parameters $\vec{y}$ by solving a single deterministic problem of size JK

  – in particular

    - one does not have to explicitly sample the random parameters $\vec{y}$

    - one does not have to determine multiple solutions of the SPDE

[†]Economies are possible for linear SPDEs; more on this later

- Note that, once the c_{jk}'s are determined, one has obtained the explicit formula

$$u_{JK}(\mathbf{x};\vec{y}) = \sum_{j=1}^{J}\sum_{k=1}^{K} c_{jk}\phi_j(\mathbf{x})\psi_k(\vec{y})$$

  for the approximate solution of the SPDE that can be evaluated at any point $\mathbf{x} \in \mathcal{D}$ in the spatial domain and for any value $\vec{y} \in \Gamma$ of the random parameters

  - in particular, one can determine, by straightforward evaluation, $u_{JK}(\mathbf{x}, \vec{y}_q)$ at any quadrature point $\vec{y}_q$ appearing in a quadrature rule approximation of a quantity of interest

- Thus, we obtain the stochastic Galerkin approximation to the quantity of interest

$$\int_{\Gamma} G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u(\mathbf{x};\vec{y}_q)\Big)$$

$$\approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_{JK}(\mathbf{x};\vec{y}_q)\Big)$$

$$= \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(\sum_{j=1}^{J}\sum_{k=1}^{K} c_{jk}\phi_j(\mathbf{x})\psi_k(\vec{y}_q)\Big)$$

- To complete the description of the problem actually solved on a computer, one has to make specific choices[†]
  - for an approximating subspace $Z_K \subset Z$
  - for a basis $\{\psi_k(\vec{y})\}_{k=1}^K$ for Z_K
  - for a quadrature rule $\{\widehat{w}_r, \widehat{\vec{y}}_r\}_R$ used to approximate the parameter integrals in the discretized SPDE
  - for a quadrature rule $\{w_q, \vec{y}_q\}_{q=1}^Q$ used to approximate the parameter integrals in the discretized quantity of interest

- We arrange our discussion according to the first two choices
  - for each choice for the approximating space and the basis set, we will make choices for the two quadrature rules

---

[†]We assume that the approximating subspace $S_J \subset S$ and a basis $\{\phi_j(\mathbf{x})\}_{j=1}^J$ used for spatial discretization have been already chosen

- For parameter approximating spaces Z_K, one can use
  - locally-supported piecewise polynomial spaces
    - i.e., a finite element-type method
  - globally-supported polynomial spaces
    - i.e., a spectral-type method

- Following this plan will enable us to show that many (if not all) numerical methods for SPDEs can be derived from the stochastic Galerkin framework

GLOBAL POLYNOMIAL APPROXIMATING SPACES – POLYNOMIAL CHAOS AND LAGRANGE INTERPOLATORY METHODS

GLOBAL POLYNOMIAL APPROXIMATING SPACES FOR PARAMETER APPROXIMATION

- Let P_r denote the set of all polynomials of degree less than or equal to r

- Let $\{\Theta_i(y)\}_{i=0}^r$ denote a basis for P_r

  – of course, there are an infinite number of possible bases

    - the simplest is the monomial basis for which
      $\Theta_i(y) = y^i$ for $i = 0, 1, \ldots, r$

  – we will discuss several bases later

- Let $p = (p_1, p_2, \ldots, p_N)$ be a multi-index, i.e.,

  – an N-vector whose components are non-negative integers

  and let $|p| = \sum_{n=1}^N p_n$

- For each parameter y_n, we use polynomials of degree M and a basis $\{\Theta_{n,k}(y_n)\}_{k=1}^{K_n}$
  - for the sake of simplicity, we assume that $M_n = M$ for all n
  - there are good reasons for sometimes choosing different degree polynomials for each parameter
    - we will point out some instances for which this is the case

- For a given integer $M \geq 0$, let $\{\psi_k(\vec{y})\}_{k=1}^{K}$ denote the set of distinct multivariate polynomials such that

$$\left\{\psi_k(\vec{y})\right\}_{k=1}^{K} = \left\{\prod_{n=1}^{N} \Theta_{n,i_n}(y_n)\right\}$$

where

$$\Theta_{n,i_n}(y_n) \in P_M \qquad \text{and} \qquad |p| \leq M$$

– the highest degree term in any of the multivariate polynomials is M

- thus, if $N = 2$ and $M = 2$, we have terms like y_1^2 and $y_1 y_2$ but not terms like $y_1^2 y_2$

– the number of probabilistic degrees of freedom is given by

$$K = \frac{(N+M)!}{N!\,M!}$$

where $N =$ number of random parameters

$M =$ maximal degree of any of the N-dimensional global poloynomials used

– for example, if $N = 2$ and $M = 3$, we have

$$|p| = p_1 + p_2 \leq M = 3$$

and

$$K = \frac{(N+M)!}{N!\,M!} = \frac{(2+3)!}{2!\,3!} = 10$$

and we have the set of 10 basis functions

$$\left\{ \psi_1(y_1, y_2) \,,\, \ldots \,,\, \psi_{10}(y_1, y_2) \right\} = \left\{ \begin{array}{c} \Theta_{1,0}(y_1)\,\Theta_{2,0}(y_2) \\ \Theta_{1,1}(y_1)\,\Theta_{2,0}(y_2) \\ \Theta_{1,0}(y_1)\,\Theta_{2,1}(y_2) \\ \Theta_{1,1}(y_1)\,\Theta_{2,1}(y_2) \\ \Theta_{1,2}(y_1)\,\Theta_{2,0}(y_2) \\ \Theta_{1,0}(y_1)\,\Theta_{2,2}(y_2) \\ \Theta_{1,2}(y_1)\,\Theta_{2,1}(y_2) \\ \Theta_{1,1}(y_1)\,\Theta_{2,2}(y_2) \\ \Theta_{1,3}(y_1)\,\Theta_{2,0}(y_2) \\ \Theta_{1,0}(y_1)\,\Theta_{2,3}(y_2) \end{array} \right\}$$

- Alternately, one could use the tensor product basis

$$\left\{\psi_k(\vec{y})\right\}_{k=1}^{K} = \left\{\prod_{n=1}^{N} \Theta_{n,i_n}(y_n)\right\}$$

where

$$\Theta_{n,i_n}(y_n) \in P_M \qquad \text{and} \qquad p_n \leq M \text{ for all } n$$

– now the highest degree term in any of the polynomials is M in each y_n

- thus, if $M = 2$, we have not only have terms like y_1^2 and $y_1 y_2$, but we also have terms like $y_1^2 y_2$ and $y_1^2 y_2^2$

– the number of probabilistic degrees of freedom is now given by

$$K = (M+1)^N$$

where $N =$ number of random parameters

$M =$ maximal degree in any variable y_n of any of the N-dimensional global poloynomials used

– for example, if $N = 2$ and $M = 3$, we have

$$K = (M+1)^N = (3+1)^2 = 16$$

$$\left\{ \psi_1(y_1, y_2) \, , \, \ldots \, , \, \psi_{16}(y_1, y_2) \right\} = \left\{ \begin{array}{l} \Theta_{1,0}(y_1) \, \Theta_{2,0}(y_2) \\ \Theta_{1,1}(y_1) \, \Theta_{2,0}(y_2) \\ \Theta_{1,0}(y_1) \, \Theta_{2,1}(y_2) \\ \Theta_{1,1}(y_1) \, \Theta_{2,1}(y_2) \\ \Theta_{1,2}(y_1) \, \Theta_{2,0}(y_2) \\ \Theta_{1,0}(y_1) \, \Theta_{2,2}(y_2) \\ \Theta_{1,2}(y_1) \, \Theta_{2,1}(y_2) \\ \Theta_{1,1}(y_1) \, \Theta_{2,2}(y_2) \\ \Theta_{1,3}(y_1) \, \Theta_{2,0}(y_2) \\ \Theta_{1,0}(y_1) \, \Theta_{2,3}(y_2) \\ \Theta_{1,1}(y_1) \, \Theta_{2,3}(y_2) \\ \Theta_{1,2}(y_1) \, \Theta_{2,3}(y_2) \\ \Theta_{1,3}(y_1) \, \Theta_{2,3}(y_2) \\ \Theta_{1,2}(y_1) \, \Theta_{2,2}(y_2) \\ \Theta_{1,3}(y_1) \, \Theta_{2,1}(y_2) \\ \Theta_{1,3}(y_1) \, \Theta_{2,2}(y_2) \end{array} \right\}$$

Global polynomial approximation in parameter space

$N =$ no. random parameters	$M =$ maximal degree of polynomials	$K =$ no. of probabilistic degrees of freedom	
		using complete polynomial basis	using tensor product basis
3	3	20	64
	5	56	216
5	3	56	1,024
	5	252	7,776
10	3	286	1,048,576
	5	3,003	60,046,176
20	3	1,771	$> 1\times10^{12}$
	5	53,130	$> 3\times10^{15}$
100	3	176,851	$> 1\times10^{60}$
	5	96,560,646	$> 6\times10^{77}$

- It seems that using tensor product bases is a bad idea

- Once a basis set $\{\psi_k(\vec{y})\}_{k=1}^K$ is chosen, we use the approximation

$$u_{J,K} = \sum_{j=1}^{J} \sum_{k=1}^{K} c_{j,k} \phi_j(\mathbf{x}) \psi_k(\vec{y})$$

  - the probabilistic basis functions $\{\psi_k(\vec{y})\}_{k=1}^K$ are multivariate global polynomials

- The discrete system involves JK equations in JK unknowns, where

  $J =$ the number of finite element degrees of freedom
  used to discretize in physical space

  $K =$ the number of global polynomials
  used to discretize in parameter space

GLOBAL ORTHOGONAL POLYNOMIAL BASES

- For $n = 1, \ldots, N$, let $\{H_{n,m_n}(y_n)\}_{m_n=0}^{M}$ denote the set of polynomials in $\mathbb{R}$ of degree less than or equal to M that are orthonormal with respect to the function $\rho_n(y_n)$

  - we have that

$$\int_{\mathcal{I}_n} H_{n,m_n}(y_n) H_{n,m'_n}(y_n) \rho_n(y_n)\, dy_n = \delta_{mm'} \qquad \text{for } m_n, m'_n \in \{0, \ldots, M\}$$

  - note that the set $\{H_{n,m_n}(y_n)\}_{m_n=0}^{M}$ is hierarchical in the sense that

$$\text{degree}(H_{n,m_n}) = m_n$$

- Let

$$\Psi_k(\vec{y}) = \prod_{n=1}^{N} H_{n,m_n}(y_n) \quad \text{for all } m_n \in \{0, \ldots, M\} \text{ such that } \sum_{n=1}^{N} m_n \leq M$$

- We then have that $k \in \left\{1, \ldots, K_{PC} = \frac{(N+M)!}{N!M!}\right\}$

- For example, if $M = 1$ and $N = 3$ we have the $K_{PC} = 4$ basis functions[†]

$$H_{1,0}(y_1)H_{2,0}(y_2)H_{3,0}(y_3)$$

$$H_{1,1}(y_1)H_{2,0}(y_2)H_{3,0}(y_3) \quad H_{1,0}(y_1)H_{2,1}(y_2)H_{3,0}(y_3) \quad H_{1,0}(y_1)H_{2,0}(y_2)H_{3,1}(y_3)$$

while for if $M = 2$ and $N = 3$ we have the $K_{PC} = 10$ basis functions (suppressing noting the explicit dependences on the $\vec{y}_n$'s)

$$H_{1,0}H_{2,0}H_{3,0}$$

$$H_{1,1}H_{2,0}H_{3,0} \quad H_{1,0}H_{2,1}H_{3,0} \quad H_{1,0}H_{2,0}H_{3,1}$$

$$H_{1,2}H_{2,0}H_{3,0} \quad H_{1,1}H_{2,1}H_{3,0} \quad H_{1,1}H_{2,0}H_{3,1} \quad H_{1,0}H_{2,2}H_{3,0} \quad H_{1,0}H_{2,1}H_{3,1} \quad H_{1,0}H_{2,0}H_{3,2}$$

[†]It is convenient to write the N-dimensional polynomials so that each row contains the polynomials of the same total degree $\sum_{n=1}^{N} m_n$; thus the first row contains all possible products of the N one-dimensional polynomials of total degree 0, the second row has total degree 1, etc.

- We see that the functions $\Psi_k(\vec{y})$'s are products of one-dimensional orthonormal polynomials and have total degree less than or equal to M
  - we then have that

$$\int_\Gamma \Psi_k(\vec{y})\Psi_{k'}(\vec{y})\rho(\vec{y})\,d\vec{y} = \int_\Gamma \Psi_k(\vec{y})\Psi_{k'}(\vec{y})\Pi_{n=1}^N \rho_n(y_n)\,d\vec{y}$$
$$= \prod_{n=1}^N \int_{\mathcal{I}_n} H_{n,m_n}(y_n)H_{n,m'_n}(y_n)\rho_n(y_n)\,dy_n = \delta_{kk'}$$

  - note that we need to write $\rho(\vec{y}) = \prod_{n=1}^N \rho_n(y_n)$, i.e., as a product as well, so that we know what $H_{n,m}(\cdot)$ is orthonormal with respect to
  - thus, we are restricted to independent random variables and to parameter domains Γ that are (possibly infinite) hypercubes

- It is easy to see that the set $\{\Psi_k\}_{k=1}^{K_{PC}}$ of N-dimensional polynomials is a basis for the complete polynomial space of degree M, i.e.,

$$\text{span}\{\Psi_k\}_{k=1}^{K_{PC}} = \text{all polynomials of total degree} \leq M$$

- The stochastic Galerkin-global orthogonal polynomial approximation of the solution of the SPDE is then defined by setting

$$Z_{PC} = \text{span}\{\Psi_k\}_{k=1}^{K_{PC}}$$

so that

$$u_{PC}(\mathbf{x}, \vec{y}) = \sum_{j=1}^{J} \sum_{k=1}^{K_{PC}} c_{jk} \phi_j(\mathbf{x}) \Psi_k(\vec{y})$$

- This is better known under another name[†]

(stochastic Galerkin) polynomial chaos approximation (SG-PC) = complete, global orthonormal polynomial approximation

[†]Polynomial chaos approximations usually refer to the case for which, for all n, $\rho_n(y_n)$ is a Gaussian PDF so that, for all n, $\{H_{n,m}(y_n)\}_{m=0}^{M}$ are sets of Hermite polynomials; for other PDFs, the SC-PC approximation is usually referred to as a generalized polynomial chaos approximation; here we do not differentiate between the two and refer to all cases as polynomial chaos approximations

- The implementation of the SG-PC method is simpler if one instead uses a tensor product polynomial space; however, as we have seen, such a choice leads to hugely more costly approximations[†]

[†]The tensor product basis is given by

$$\Psi_k(\vec{y}) = \prod_{n=1}^{N} H_{n,m_n}(y_n) \quad \text{for all } m_n \in \{0,\ldots,M\} \text{ such that } m_n \leq M$$

in this case, $\text{span}\{\Psi_k\}_{k=1}^{K}$ is the tensor product space of polynomials such that the degree in any coordinate y_n is less than or equal to M; if we do this, we end up with $K = (M+1)^N$ basis functions; for example, if $M = 1$ and $N = 3$, we have the 8 polynomials (the 4 we had before plus 4 additional ones)

$$\begin{array}{ccc} & H_{1,0}H_{2,0}H_{3,0} & \\ H_{1,1}H_{2,0}H_{3,0} & H_{1,0}H_{2,1}H_{3,0} & H_{1,0}H_{2,0}H_{3,1} \\ H_{1,1}H_{2,1}H_{3,0} & H_{1,1}H_{2,0}H_{3,1} & H_{1,0}H_{2,1}H_{3,1} \\ & H_{1,1}H_{2,1}H_{3,1} & \end{array}$$

for $N > 1$ and $M > 0$ we have that $(M+1)^N > \frac{(N+M)!}{N!M!}$; for a moderate number of parameters or for a moderately high degree polynomial, we in fact have that $(M+1)^N \gg \frac{(N+M)!}{N!M!}$; for example,

$$\text{if } M = 6 \text{ and } N = 3 \Longrightarrow (N+M)!/(N!M!) = 84 \text{ and } (M+1)^N = 343$$
$$\text{if } M = 4 \text{ and } N = 5 \Longrightarrow (N+M)!/(N!M!) = 126 \text{ and } (M+1)^N = 3125$$
$$\text{if } M = 2 \text{ and } N = 7 \Longrightarrow (N+M)!/(N!M!) = 36 \text{ and } (M+1)^N = 2187$$

the disparity gets worse as, say, N increases; for example,

$$\text{if } M = 2 \text{ and } N = 10 \Longrightarrow (N+M)!/(N!M!) = 66 \text{ and } (M+1)^N = 59059$$

on the other hand, since the accuracy, i.e., the rate of convergence of global polynomial approximation, is determined by the degree of the largest complete polynomial space contained in the approximate space, for the same M, the accuracy obtained using a tensor product space is the same as that obtained using a complete polynomial space; as a result, by using the latter one can obtain the same accuracy with substantially fewer degrees of freedom

SG-PC approximations of quantities of interest

- The SG-PC approximation of a quantity of interest is then defined by

$$\int_{\Gamma} G\big(u(\mathbf{x};\vec{y})\big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\big(u_{PC}(\mathbf{x};\vec{y}_q)\big)$$

where

- $u_{PC}(\mathbf{x};\vec{y}_q)$, $q = 1,\ldots,Q$, is obtained by evaluation of the SG-PC approximation of the stochastic SPDE at the quadrature points
  - i.e., we have that

$$u_{PC}(\mathbf{x},\vec{y}_q) = \sum_{j=1}^{J}\sum_{k=1}^{K_{PC}} c_{jk}\phi_j(\mathbf{x})\Psi_k(\vec{y}_q) \qquad \text{for } q = 1,\ldots,Q$$

- Thus, the SG-PC approximation of a quantity of interest can be determined by

  1. first solving a single $JK_{PC} \times JK_{PC}$ system of equations to determine the SG-PC approximation of the solution of the SPDE;

  2. then evaluating the SG-PC approximation at the Q quadrature points;

  3. substituting the results of Step 2 into the quadrature rule approximation of the quantity of interest

- The cost of obtaining an SG-PC approximation of a quantity of interest is dominated by the first step

GLOBAL LAGRANGE INTERPOLATORY BASES

- Instead of using global orthogonal polynomials to define a stochastic Galerkin method, one can use interpolatory polynomials

- Given a set of points $\{\widetilde{\vec{y}}_k\}_{k=1}^{K_{LI}}$ in Γ

  - for $k \in \{1, \ldots, K_{LI}\}$, let $L_k(\vec{y})$ denote the set of Lagrange interpolating polynomials for these points

    - we have that

$$L_k(\vec{y}_{k'}) = \delta_{kk'} \qquad \text{for all } k, k' \in \{1, \ldots, K_{LI}\}$$

- Set $\psi_k(\vec{y}) = L_k(\vec{y})$ for $k \in \{1, \ldots, K_{LI}\}$ so that

$$Z_{K_{LI}} = \text{span}\{L_k\}_{k=1}^{K_{LI}}$$

- Then, the stochastic Galerkin-Lagrange interpolant (SG-LI) approximation of the solution of the SPDE takes the form

$$u_{LI}(\mathbf{x}, \vec{y}) = \sum_{j=1}^{J} \sum_{k=1}^{K_{LI}} c_{jk} \phi_j(\mathbf{x}) L_k(\vec{y})$$

- In general, the SG-LI approximation to the solution of an SPDE can be obtained by solving a single $JK_{LI} \times JK_{LI}$ system
  - this would also be the dominant cost encountered in obtaining an SG-LI approximation of a quantity of interest

- If we choose a point set $\{\widetilde{\vec{y}}_k\}_{k=1}^{K_{LI}}$ that can be used to define a complete interpolating polynomial of degree less than or equal M, we have that

$$Z_{K_{LI}} = Z_{K_{PC}} \qquad \text{and} \qquad K_{LI} = K_{PC} = \frac{(N+M)!}{N!M!}$$

- In this case, it is clear that

the polynomial chaos approximation $u_{PC}(\mathbf{x};\vec{y})$
= global Lagrange interpolant approximation
$u_{LI}(\mathbf{x};\vec{y})$ based on a complete polynomial space

  - the only differences between the two approximations result from the choices of bases

- Unfortunately, even for a moderate number of parameters, it may not be easy to define a "good" set of interpolation points that can be used to determine a complete Lagrange interpolant
  - it is easy to define a set of interpolation points that can be used to define a tensor product Lagrange interpolant[†]
  - however, as we have seen, this leads to a very inefficient approximation compared to complete polynomial approximation

- There exists intermediate choices, e.g., Smolyak point sets, that can be systematically defined in any dimension
  - for the Smolyak point sets, $K_{LI} > \dfrac{(M+N)!}{N!M!}$ so that they require more points compared to complete polynomial interpolation
  - however, we have that $K_{LI} \ll (M+1)^N$ so that it requires much fewer points compared to tensor product interpolation

[†]Unlike the case for orthogonal polynomials, for Lagrange polynomials it is not easy to define a complete polynomial basis from the tensor product basis; for the Lagrange case, the tensor product basis is not hierarchical since all Lagrange polynomials are of the same degree

- We therefore conclude that

in general, for the same accuracy, a stochastic Galerkin-Lagrange polynomial approximation is (a little) more costly to obtain than is a stochastic Galerkin-polynomial chaos approximation

- However, as we shall now see, a judicious choice for the interpolation points can lead to great efficiency improvements in stochastic Galerkin-Lagrange interpolation methods

  – we defer discussion of how one one obtains the LI-approximation of a quantity of interest until after we consider this special case of the SG-LI method

  – we also defer further discussion of Smolyak point sets until later

STOCHASTIC COLLOCATION METHODS

- For the SG-LI method, the discretized SPDE looks like

$$\sum_{r=1}^{R} \widehat{w}_r \rho(\widehat{\vec{y}}_r) L_{k'}(\widehat{\vec{y}}_r) \int_{\mathcal{D}} S\Big(\sum_{j=1}^{J}\sum_{k=1}^{K} c_{jk}\phi_j(\mathbf{x}) L_k(\widehat{\vec{y}}_r), \widehat{\vec{y}}_r\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x}$$

$$= \sum_{r=1}^{R} \widehat{w}_r \rho(\widehat{\vec{y}}_r) L_{k'}(\widehat{\vec{y}}_r) \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\widehat{\vec{y}}_r)\, d\mathbf{x}$$

for $j' \in \{1, \ldots, J\}$ and $k' \in \{1, \ldots, K\}$

- Suppose we choose

the interpolating points $\{\widetilde{\vec{y}}_k\}_{k=1}^{K_{LI}}$ for the SG-LI method

to be the same as

the quadrature points $\{\widehat{\vec{y}}_r\}_{r=1}^{R}$ used in the discretized SPDE

- We then have that

$$L_k(\widehat{\vec{y}}_r) = \delta_{kr} \qquad \forall\, r, k \in \{1, \ldots, R = K_{LI}\}$$

- As a result, the discretized SPDE reduces to

$$\int_{\mathcal{D}} S\Big(\sum_{j=1}^{J} c_{jr}\phi_j(\mathbf{x}), \widehat{\vec{y}}_r\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\widehat{\vec{y}}_r)\, d\mathbf{x}$$

$$\text{for } j' \in \{1, \ldots, J\},\ r \in \{1, \ldots, R = K_{LI}\}$$

- Thus, we have total uncoupling in parameter space

  – for each $r \in \{1, \ldots, R\}$, we can solve the separate standard, deterministic finite element problem for $\{c_{jr}\}_{j=1}^{J}$

for $r \in \{1, \ldots, R\}$, determine $u_r(\mathbf{x}) = \sum_{j=1}^{J} c_{jr}\phi_j(\mathbf{x})$ satisfying

$$\int_{\mathcal{D}} S\Big(u_r(\mathbf{x}), \widehat{\vec{y}}_r\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\widehat{\vec{y}}_r)\, d\mathbf{x}$$

$$\text{for } j' \in \{1, \ldots, J\}$$

- Such a method is referred to as a stochastic collocation (SC) method so that

stochastic collocation methods are
stochastic Galerkin-Lagrange interpolation methods for which
the interpolation points are the same as the quadrature points
of the quadrature rule used to discretize the SPDE

- It is important to note that for stochastic collocation methods,

the uncoupling of the spatial and probabilistic degrees of freedom

occurs for

general nonlinear PDEs

general joint probability distributions

and

general random field data

- If desired, the stochastic collocation approximation to the solution $u(\mathbf{x}, \vec{y})$ of the SPDE is then given by

$$u_{SC}(\mathbf{x}, \vec{y}) = \sum_{r=1}^{R} u_r(\mathbf{x}) L_r(\vec{y}) = \sum_{j=1}^{J} \sum_{r=1}^{R} c_{jr} \phi_j(\mathbf{x}) L_r(\vec{y})$$

  - however, as we will now see, one does not need to form this expression to a determine an approximation of a quantity of interest

  - this is unlike the case for general stochastic Galerkin methods, including polynomial chaos methods, for which one must evaluate the approximation of the solution of the SPDE at the quadrature points of the approximation of a quantity of interest

SC-approximations of quantities of interest

- It is also convenient to use the same quadrature rule

  - to approximate a quantity of interest

  as was used to

  - approximate the integrals in the discretized SPDE

  and that was also used as

  - the Lagrange interpolations points,

  i.e., we choose

$$K_{LI} = R = Q$$

$$\{\widetilde{\vec{y}}_k\}_{k=1}^{K_{LI}} = \{\widehat{\vec{y}}_r\}_{r=1}^{R} = \{\vec{y}_q\}_{q=1}^{Q} \qquad \text{and} \qquad \{\widehat{w}_r\}_{r=1}^{R} = \{w_q\}_{q=1}^{Q}$$

- We then have that

$$L_r(\vec{y}_q) = \delta_{rq} \qquad \text{for all } r, q \in \{1, \ldots, K_{LI} = R = Q\}$$

- Using this in the expression for the approximation of a quantity of interest results in

$$\int_{\Gamma} G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_{SC}(\mathbf{x})\Big)$$

$$= \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(\sum_{r=1}^{R} u_r(\mathbf{x}) L_r(\vec{y}_q)\Big) = \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_q(\mathbf{x})\Big)$$

i.e.,

$$\int_{\Gamma} G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_q(\mathbf{x})\Big)$$

where, for $q \in \{1,\ldots,Q = R = K_{LI}\}$, $u_q(\mathbf{x}) = \sum_{j=1}^{J} c_{jq}\phi_j(\mathbf{x})$
is determined from

$$\int_{\mathcal{D}} S\Big(u_q(\mathbf{x}),\vec{y}_q\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\,d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\vec{y}_q)\,d\mathbf{x} \qquad \text{for } j' \in \{1,\ldots,J\}$$

- Note that
  - we do not have to explicitly determine the Lagrange interpolating polynomials $\{L_k(\vec{y})\}_{k=1}^{K_{LI}}$ to determine the approximation of a quantity of interest
  - nor do we have to form and evaluate, at quadrature points, the SC-approximation[†]

- Thus, we see that the SC-approximation of a quantity of interest can be determined by
  1. first solving $Q = K_{LI}$ systems of equations of size J to determine $u_q(\mathbf{x})$ for $q = 1, \ldots, Q = K_{LI}$;
  2. then substituting the results of Step 1 into the approximation of the quantity of interest

[†]In contrast, for PC approximations of quantities of interest one must explicitly evaluate the PC approximation at quadrature points

- The cost of obtaining the SC-approximation of a quantity of interest is dominated by the first step which requires the solution of K_{LI} systems of size J

  - recall that the cost of obtaining the PC-approximation of a quantity of interest is dominated by the cost of solving a single deterministic system of size JK_{PC}

  - for general, nonlinear problems, the SC-approximation can be obtained at much less cost[†]

[†]In the best-case scenario for which the PC-system of size JK_{PC} and each of the $Q = R = K_{LI}$ SC-systems of size J can be solved in linear time, the solution cost associated with the PC-approximation of a quantity of interest will be of $O(JK_{PC})$ while the corresponding solution cost for the SC-approximation of a quantity of interest is of $O(JK_{LI})$; for the same accuracy, in practice $K_{LI} > K_{PC}$ so that in this best-case scenario, the SC-approximation of a quantity of interest is more costly to obtain than is the PC-approximation; for more general problems for which solution costs are not linear in the number of degrees of freedom, the PC-approximation is more costly to obtain that is the SC-approximation since, for some $\alpha > 1$, one must compare the cost of $O(JK_{PC})^{\alpha}$ for the PC case to the cost of $O(J^{\alpha}K_{LI})$ for the SC case, keeping in mind that although $K_{LI} > K_{PC}$, using Smolyak points as collocation points we have that $K_{LI} \approx K_{PC}$

NON-INTRUSIVE POLYNOMIAL CHAOS METHODS

- Can the uncoupling of parameter and spatial degrees of freedom be effected in a polynomial chaos setting?

- The PC approximation is given by

$$u_{PC}(\mathbf{x}, \vec{y}) = \sum_{j=1}^{J} \sum_{k=1}^{K_{PC}} c_{jk} \phi_j(\mathbf{x}) \Psi_k(\vec{y}) = \sum_{k=1}^{K_{PC}} \widetilde{u}_k(\mathbf{x}) \Psi_k(\vec{y})$$

  where for $k \in \{1, \ldots, K_{PC}\}$,

$$\widetilde{u}_k(\mathbf{x}) = \sum_{j=1}^{J} c_{jk} \phi_j(\mathbf{x})$$

  and $\{\Psi_k(\vec{y})\}_{k=1}^{K_{PC}}$ is a set of orthonormal polynomials with respect to weight $\rho(\vec{y}) = \prod_{n=1}^{N} \rho_n(y_n)$

- As a result, we have that, for $k' \in \{1, \ldots, K_{PC}\}$,

$$\int_{\Gamma} u_{PC}(\mathbf{x}, \vec{y}) \Psi_{k'}(\vec{y}) \rho(\vec{y}) \, d\vec{y} = \sum_{k=1}^{K_{PC}} u_k(\mathbf{x}) \int_{\Gamma} \Psi_k(\vec{y}) \Psi_{k'}(\vec{y}) \rho(\vec{y}) \, d\vec{y} = \widetilde{u}_{k'}(\mathbf{x})$$

- We view this as a formula for $\widetilde{u}_{k'}(\mathbf{x})$, i.e.,

$$\widetilde{u}_{k'}(\mathbf{x}) = \sum_{j=1}^{J} c_{jk'}\phi_j(\mathbf{x}) = \int_{\Gamma} u_{PC}(\mathbf{x}, \vec{y})\Psi_{k'}(\vec{y})\rho(\vec{y})\, d\vec{y}$$

- We use a quadrature rule[†] $\{\widehat{w}_r, \widehat{\vec{y}}_r\}_{r=1}^{R}$ to approximate the integral to obtain

$$\widetilde{u}_{k'}(\mathbf{x}) \approx \sum_{r=1}^{R} \widehat{w}_r u_{PC}(\mathbf{x}, \widehat{\vec{y}}_r)\Psi_{k'}(\widehat{\vec{y}}_r)\rho(\widehat{\vec{y}}_r) \qquad \text{for } k' \in \{1, \ldots, K_{PC}\}$$

- For $r \in \{1, \ldots, R\}$, we replace $u_{PC}(\mathbf{x}, \widehat{\vec{y}}_r)$ by the solution $u_r(\mathbf{x})$ of[‡]

$$\int_{\mathcal{D}} S\Big(u_r(\mathbf{x}), \widehat{\vec{y}}_r\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\widehat{\vec{y}}_r)\, d\mathbf{x} \qquad \text{for } j' \in \{1, \ldots, J\}$$

[†] This quadrature rule may be the same or may be different from the quadrature rule used to approximate a quantity of interest

[‡] Note that this is exactly the same set of R equations that is solved for in the stochastic collocation case

- We thus obtain

$$\widetilde{u}_{k'}(\mathbf{x}) \approx \sum_{r=1}^{R} \widehat{w}_r u_r(\mathbf{x}) \Psi_{k'}(\widehat{\vec{y}}_r) \rho(\widehat{\vec{y}}_r)$$

- We use this approximation to define the[†] non-intrusive polynomial chaos (NIPC) approximation to the solution $u(\mathbf{x}, \vec{y})$ of the SPDE:[‡]

$$u(\mathbf{x}, \vec{y}) \approx u_{PC}(\mathbf{x}, \vec{y}) = \sum_{k=1}^{K_{PC}} \widetilde{u}_k(\mathbf{x}) \Psi_k(\vec{y})$$

$$\approx u_{NIPC}(\mathbf{x}, \vec{y}) = \sum_{k=1}^{K_{PC}} \sum_{r=1}^{R} \widehat{w}_r u_r(\mathbf{x}) \Psi_k(\widehat{\vec{y}}_r) \rho(\widehat{\vec{y}}_r) \Psi_k(\vec{y})$$

[†] Nowadays, the polynomial chaos method previously discussed is often referred as the intrusive polynomial chaos method to differentiate it from the non-intrusive polynomial chaos method defined here

[‡] In comparison, the stochastic collocation approximation takes the simpler form

$$u_{SC}(\mathbf{x}, \vec{y}) = \sum_{r=1}^{R} u_r(\mathbf{x}) L_r(\vec{y})$$

due to the fact that $L_k(\widehat{\vec{y}}_r) = \delta_{kr}$ in the SC case while $\Psi_k(\widehat{\vec{y}}_r) \neq 0$ for all k and r in the NIPC case

- Thus, the NIPC approximation can be obtained by solving

  R deterministic problems of size J to obtain $u_r(\mathbf{x})$ for $r = 1, \ldots, R$

  instead of the

  one deterministic problem of size JK_{PC}

  that is solved in the intrusive polynomial chaos method

- All K_{PC} "coefficients" $\sum_{r=1}^{R} \widehat{w}_r u_r(\mathbf{x}) \Psi_k(\widehat{\vec{y}}_r) \rho(\widehat{\vec{y}}_r)$, $k \in \{1, \ldots, K_{PC}\}$, in the NIPC expansion

$$
\begin{aligned}
u_{NIPC}(\mathbf{x}, \vec{y}) &= \sum_{k=1}^{K_{PC}} \sum_{r=1}^{R} \widehat{w}_r u_r(\mathbf{x}) \Psi_k(\widehat{\vec{y}}_r) \rho(\widehat{\vec{y}}_r) \Psi_k(\vec{y}) \\
&= \sum_{r=1}^{R} \widehat{w}_r \rho(\widehat{\vec{y}}_r) u_r(\mathbf{x}) \sum_{k=1}^{K_{PC}} \Psi_k(\widehat{\vec{y}}_r) \Psi_k(\vec{y})
\end{aligned}
$$

  can be obtained from the same R solutions $u_r(\mathbf{x})$, $r \in \{1, \ldots, R\}$, of the SPDE

- The cost of obtaining the NIPC-approximation is dominated by the need to solve[†] R systems of size J

- For non-intrusive-polynomial chaos approximations,

  the uncoupling of the spatial and probabilistic degrees of freedom

  occurs for

  general nonlinear PDEs

  but only for

  independent random variables[‡]

  and

  Gaussian random field data[‡]

[†] This is just the same as for the stochastic collocation approximation

[‡] This is unlike the case for stochastic collocation methods for which similar uncouplings are possible for general joint probability distributions and general random fields

- Thus, it is clear that

non-intrusive polynomial chaos approximations are stochastic Galerkin-global orthogonal polynomial approximations obtained by approximating the coefficients of the orthogonal polynomials via a quadrature rule

- It is also clear that, for the same accuracy

the costs of obtaining stochastic collocation and non-intrusive polynomial chaos approximations are comparable and, in general, both are much lower than the cost of obtaining the intrusive polynomial chaos approximation

NIPC-approximations of quantities of interest

- Unlike the stochastic collocation case, there is no great advantage to using the same quadrature rule for approximating a quantity of interest as is used to construct the non-intrusive polynomial chaos approximation

  – on the other hand, there is no reason not to do so

  – so, we choose

$$Q = R, \qquad \{w_q\}_{q=1}^{Q} = \{\widehat{w}_r\}_{r=1}^{R}, \qquad \text{and} \qquad \{\vec{y}_q\}_{q=1}^{Q} = \{\widehat{\vec{y}}_r\}_{r=1}^{R}$$

- Then, the NIPC approximation of a quantity of interest has the form[†]

$$\int_\Gamma G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_{NIPC}(\mathbf{x})\Big)$$

$$= \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Bigg(\sum_{k=1}^{K_{PC}} \Bigg(\sum_{q=1}^{Q} w_q u_q(\mathbf{x}) \Psi_k(\vec{y}_q)\rho(\vec{y}_q)\Bigg)\Psi_k(\vec{y}_q)\Bigg)$$

where, for $q \in \{1,\ldots,Q=R\}$, $u_q(\mathbf{x}) = \sum_{j=1}^{J} c_{jq}\phi_j(\mathbf{x})$ is determined from

$$\int_{\mathcal{D}} S\Big(u_q(\mathbf{x}),\vec{y}_q\Big)T\Big(\phi_{j'}(\mathbf{x})\Big)\,d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\vec{y}_q)\,d\mathbf{x} \qquad \text{for } j' \in \{1,\ldots,J\}$$

[†] In comparison, the stochastic collocation approximation of the quantity of interest takes the simpler form

$$\int_\Gamma G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_q(\mathbf{x})\Big)$$

again due to the fact that $L_k(\widehat{\vec{y}}_r) = \delta_{kr}$ in the SC case while $\Psi_k(\widehat{\vec{y}}_r) \neq 0$ for all k and r in the NIPC case

- Thus, we see that the NIPC approximation of a quantity of interest can be determined by

  1. first solving Q systems of equations of size J to determine $u_q(\mathbf{x})$ for $q = 1, \ldots, Q$;
  2. then substituting the results of Step 1 into the NIPC-approximation of the quantity of interest

- Note that one is not restricted to use of any particular quadrature rule, either to determine the NIPC approximation of the solution of the SPDE or the NIPC approximation to a quantity of interest

  – in particular, one does not have to use interpolatory quadrature rules

  – one can use, e.g., any of the rules to be discussed in connection with stochastic sampling methods

- Note also that to obtain this approximation, one has to explicitly construct and evaluate, at the quadrature points $\vec{y}_q$, the non-intrusive polynomial chaos approximation

  - this includes having to explicitly evaluate the orthogonal polynomial basis functions $\Psi_k(\cdot)$ at the quadrature points

  - this should be contrasted with the SC approximation of a quantity of interest that does not need the explicit construction or evaluation of the SC approximation nor of the the Lagrange interpolatory polynomial basis functions $L_k(\cdot)$

  - again, these differences between the two methods are due to the facts that $L_k(\vec{y}_q) = \delta_{kq}$ while $\Psi_k(\vec{y}_q) \neq 0$ for all k and q

STOCHASTIC SAMPLING METHODS

APPROXIMATING QUANTITIES OF INTEREST USING SAMPLING METHODS

- Recall that quantities of interest often require the evaluation of stochastic integrals of functions of the solutions
- These integrals usually have to be approximated using quadrature rules, i.e.,

$$\int_\Gamma G\big(u(\mathbf{x},\vec{y});\mathbf{x},\vec{y}\big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q G\big(u(\mathbf{x},\vec{y}_q);\mathbf{x},\vec{y}_q\big)$$

or

$$\int_\Gamma G\big(u(\mathbf{x},\vec{y});\mathbf{x},\vec{y}\big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(y_q) G\big(u(\mathbf{x},\vec{y}_q);\mathbf{x},\vec{y}_q\big)$$

- To use such a rule, one needs to know the solution $u(\mathbf{x},\vec{y})$ of the SPDE at each of the quadrature points $\vec{y}_q$, $q = 1,\ldots,Q$, in the probabilistic domain Γ
  - for this purpose, one can use a stochastic Galerkin method to obtain an approximation to the the solution $u(\mathbf{x},\vec{y})$ and then evaluate that approximation at the quadrature points

- However, once a quadrature rule is chosen to approximate a quantity of interest,

  - i.e., once the quadrature points $\{\vec{y}_q\}_{q=1}^{Q}$ are known

  the simplest and most direct means of determining $u(\mathbf{x}, \vec{y}_q)$ is to simply solve the PDE Q times, once for each quadrature point $\vec{y}_q$

- This approach is referred to as the stochastic sampling method (SSM) for SPDEs and for quantities of interest that depend on the solutions of SPDEs

- We have already encountered two SSMs

  – we have seen that SGMs based on Lagrange interpolating polynomials reduce to SSMs

  – we have also seen that non-intrusive polynomial chaos methods are essentially SSMs

    - although one does need the additional step of explicitly constructing the non-intrusive polynomial chaos approximation

- In an SSM, to determine an approximation to a quantity of interest,
  - one chooses a quadrature rule for the probabilistic integrals, i.e.,
    - one chooses quadrature weights and points $\{w_q, \vec{y}_q\}_{q=1}^{Q}$
  - one chooses a finite element method, (i.e., a finite element space and a basis $\{\phi_j\}_{j=1}^{J}$ for that space) and, for each q, one defines the finite element approximation of the solution at the quadrature points by

$$u_q(\mathbf{x}) = \sum_{j=1}^{J} b_{j,q}\phi_j(\mathbf{x}) \qquad \text{for } q = 1, \ldots, Q$$

  - then, to determine $b_{j,q}$ for $j = 1, \ldots, J$ and $q = 1 \ldots, Q$, one separately, and if desired, in parallel, solves the Q deterministic problems: for $q = 1, \ldots, Q$,

$$\int_{\mathcal{D}} S\Big(\sum_{j=1}^{J} b_{j,q}\phi_j, \vec{y}_q\Big) T(\phi_{j'})\, d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'} f(\vec{y}_q)\, d\mathbf{x} \qquad \text{for } j' = 1, \ldots, J$$

- each of these can be discretized using a finite element method

$\Longrightarrow$ one can use legacy codes as black boxes

$\Longrightarrow$ i.e., without changing a single line of code

$\Longrightarrow$ i.e., one just uses the legacy code Q times

– and finally, one just substitues $u_q(\mathbf{x})$ wherever $u(\mathbf{x};\vec{y}_q)$ is needed into the quadrature rule approximation of a quantity of interest

- The cost of determining an approximation to a quantity of interest using the SSM approach is dominated by

  – the cost to determine Q finite element solutions, each of size J

- This should be compared to the cost of using general SGM approaches for the same purpose that are dominated by

  – the cost needed to determine the solution of a single system of size JK

- Which approach wins, i.e., which one yields a desired accuracy in the statistics of quantities of interest for the lowest computational cost, depends on
  - the value of Q, the number of quadrature points in SSM approaches
  - the value of K, the number of probabilistic terms in the SGM approximation to the solution
  - the cost of solving the systems of discrete equations encountered
    - for nonlinear problems and time dependent problems, one may have to solve such systems many times
  - many implementation issues

- Of course, such comparisons do not factor in the relative programming cost for implementing the different approaches
  - SSM approaches allow for the easy use of legacy codes
  - general SGM approaches do not allow for this

- In most cases, and certainly due to some recent developments, SSMs win over SGMs
  - which is why polynomial chaos people are now doing non-intrusive polynomial chaos which is, as we have seen, practically a SSM

- Of course, there are many ways to sample points in parameter space other than at the quadrature points for some integration rule
  - so, we now take a more general view of SSMs

STOCHASTIC SAMPLING METHODS ARE STOCHASTIC GALERKIN METHODS

- From the previous discussions, it seems that we could have introduced stochastic sampling methods as a special case of stochastic Galerkin methods
  - in fact,

every stochastic sampling method
is a stochastic Galerkin method using
Lagrange interpolating polynomials
based on the sample points
and quadrature rules also based on the sample points

- However, stochastic sampling methods are easier to understand through the straightforward approach we have just taken
  - the straightforward approach also avoids difficult questions about the relations of the cardinality of the set of sample points and the construction of interpolating polynomials

SURROGATE APPROXIMATIONS AND STOCHASTIC SAMPLING METHODS

- Stochastic sampling methods (SSMs) for solving stochastic PDEs are based on
  - first determining a sample set of values $\{\vec{y}_s\}_{s=1}^{N_{sample}}$ of the vector of random parameters $\vec{y} \in \Gamma \subset \mathbb{R}^N$
  - then determining N_{sample} (approximate) solutions $\{u(\mathbf{x}; \vec{y}_s)\}_{s=1}^{N_{sample}}$ of the PDE via, e.g., a finite element method

Evaluating quantities of interest within the SSM framework

- If we want to evaluate quantities of interest that involve integrals over the parameter set Γ using a Q-point quadrature rule involving the quadrature points $\{\vec{y}_q\}_{q=1}^{Q} \subset \overline{\Gamma}$ and quadrature weights $\{w_q\}_{q=1}^{Q}$
  - it is then natural to choose the set of sample points $\{\vec{y}_s\}_{s=1}^{N_{sample}}$ that are used to solve the PDE N_{sample} times to be the same as the set of quadrature points $\{\vec{y}_q\}_{q=1}^{Q}$ that are used to approximate the quantities of interest

- Alternately, we could choose $\{\vec{y}_s\}_{s=1}^{N_{sample}}$ to be different (and presumably coarser) than the quadrature points $\{\vec{y}_q\}_{q=1}^{Q}$
  - one would then use the sample points $\{\vec{y}_s\}_{s=1}^{N_{sample}}$ to build a surrogate or response surface $u_{surrogate}(\mathbf{x}, y)$ for the solution $u(\mathbf{x}, y)$
  - surrogates/response surfaces for the solution $u(\mathbf{x}, \vec{y})$ are (usually polynomial) functions of, in our case, the random parameters $\vec{y}$

– in fact, they are simply representations, e.g., in terms of Lagrange interpolation polynomials, of the approximate solution in terms of the parameter vector $\vec{y}$

– it is usually more efficient to build a surrogate/response surface directly for the integrand $G\Big(u(\mathbf{x},\vec{y});\mathbf{x},\vec{y}\Big)$ of the desired quantity of interest

- one solves for an approximation $u_s(\mathbf{x})$ to the solution $u(\mathbf{x},\vec{y}_s)$ of the PDE for the sample parameter points $\vec{y}_s$, $s=1,\ldots,N_{sample}$

- one then evaluates the approximations to the integrand

$$G_s(\mathbf{x}) = G\Big(u_s(\mathbf{x});\mathbf{x},\vec{y}_s\Big) \qquad \text{for } s=1,\ldots,N_{sample}$$

- from these samplings of G at the sample points $\vec{y}_s$, one builds a surrogate $G_{surrogate}(\mathbf{x},\vec{y})$

– once a surrogate/response surface is built, it can be used to evaluate the integrand at the quadrature points $\{\vec{y}_q\}_{q=1}^{Q}$

- To illustrate the different approaches, within the SSM framework, for computing approximations of quantities of interest, consider a quantity of the form

$$\mathcal{J}(u) = \int_{\Gamma} \int_{\mathcal{D}} G\big(u(\mathbf{x}, \vec{y})\big) \rho(\vec{y}) \, d\mathbf{x} d\vec{y}$$

  - a spatial quadrature rule with the points $\mathbf{x}_r$ and weights W_r for $r = 1, \ldots, R$ is used to approximate the spatial integral resulting in the approximation

$$\mathcal{J}(u) \approx \int_{\Gamma} \sum_{r=1}^{R} W_r G\big(u(\mathbf{x}_r, \vec{y})\big) \rho(\vec{y}) \, d\vec{y}$$

  - a parameter-space quadrature rule with the points y_q and weights w_q for $q = 1, \ldots, Q$ is used to approximate the spatial integral resulting in the approximation

$$\mathcal{J}(u) \approx \sum_{q=1}^{Q} \sum_{r=1}^{R} w_q W_r \rho(\vec{y}_q) G\big(u(\mathbf{x}_r, \vec{y}_q)\big)$$

– a set of points $\{\vec{y}_s\}_{s=1}^{N_{sample}}$ is chosen in the parameter domain Γ

- these sample points are used to obtain the set of realizations $\{u_s(\mathbf{x})\}_{s=1}^{N_{sample}}$ of a finite element discretization of the SPDE

- each realization is determined by setting the parameters $\vec{y} = \vec{y}_s$ in the discretized SPDE

– if the probalistic quadrature points $\{\vec{y}\}_{q=1}^{Q}$ are the same as the sample points $\{\vec{y}\}_{s=1}^{N_{sample}}$, we directly define the computable approximation

$$\mathcal{J}(u) \approx \sum_{q=1}^{Q} \sum_{r=1}^{R} w_q W_r \rho(\vec{y}_q) G\big(u_q(\mathbf{x}_r)\big)$$

where we have, of course, renamed $u_s(\mathbf{x})$ by $u_q(\mathbf{x})$ since now they are one and the same

– if the the sample points $\{\vec{y}\}_{s=1}^{N_{sample}}$ are coarser than the probalistic quadrature points $\{\vec{y}\}_{q=1}^{Q}$, we first build a surrogate $G_{surrogate}(\mathbf{x}_r, \vec{y})$ for $G(\mathbf{x}_r, \vec{y})$

- the simplest means for doing this is to use the set of Lagrange interpolating polynomials $\{L_s(\vec{y})\}_{s=1}^{N_{sample}}$ corresponding to the sample points $\{\vec{y}_s\}_{s=1}^{N_{sample}}$, resulting in the surrogate approximation

$$G_{surrogate}(\mathbf{x}_r, \vec{y}) = \sum_{s=1}^{N_{sample}} G\big(u_s(\mathbf{x}_r)\big) L_s(\vec{y})$$

- other surrogate constructions may be used, e.g., least-squares fits to the data $\{\vec{y}_s, G\big(u_s(\mathbf{x}_r)\big)\}_{s=1}^{N_{sample}}$ using global orthogonal polynomials or even piecewise polynomials

- once the surrogate $G_{surrogate}(\mathbf{x}_r, \vec{y})$ has been constructed,
  one defines the indirect computable approximation

$$\mathcal{J}(u) \approx \sum_{q=1}^{Q} \sum_{r=1}^{R} w_q W_r \rho(\vec{y}_q) G_{surrogate}(\mathbf{x}_r, \vec{y}_q)$$

  by evaluating the surrogate at the
  probabilistic quadrature points $\{\vec{y}_q\}_{q=1}^{Q}$

- for example, if the surrogate is constructed using
  Lagrange interpolating polynomials, we have the approximation

$$\mathcal{J}(u) \approx \sum_{s=1}^{N_{sample}} \sum_{r=1}^{R} W_r G\big(u_s(\mathbf{x}_r)\big) \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) L_s(\vec{y}_q)$$

- of course, if the sample points $\{\vec{y}_s\}_{s=1}^{N_{sample}}$ are the same as the probabilistic quadrature points $\{\vec{y}_q\}_{q=1}^{N_q}$ so that $L_s(\vec{y}_q) = \delta_{sq}$, this approximation reduces to the one obtained before which, in this example, takes the simple form

$$\mathcal{J}(u) \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) \sum_{r=1}^{R} W_r G\big(u_q(\mathbf{x}_r)\big)$$

- Note that if one uses the sample points directly as quadrature points, then one does not need to construct a representation of the approximate solution in terms of the random parameters
  - if one uses a coarser set of sampling points relative to the quadrature points, one does have to build such a representation since it needs to be evaluated at the quadrature points, and not just the sample points
  - of course, this is also unlike the case for general SGMs in which one does build such a representation, e.g., an intrusive polynomial chaos expansion

- We will concentrate on the case where the sample points are used directly as quadrature points

- So, we next discuss quadrature rules that can be used to approximate quantities of interest
  - (coarser) versions of some of these rules can also supply sample points that can be used to build surrogates or response surfaces

- We will discuss quadrature rules for the N-dimensional hypercube, the case that most often arises in practice
  - other rectangular regions, i.e., bounding boxes, can be mapped in the obvious way to the unit hypercube

- Unfortunately, we do not have time to discuss sampling in unbounded domains or in general, non-rectangular domains

QUADRATURE RULES FOR HYPERCUBES

- One is tempted to use well-known quadrature rules to define the sample points for SSMs

- We will discuss two classes of quadrature rules
  for the N-dimensional hypercube

  – sampling and simple averaging rules

    - the canonical example is Monte Carlo integration

  – weighted quadrature rules based on standard one-dimensional rules

    - ultimately, we consider sparse grid Smolyak quadrature rules

- Recall that in the SSM framework we are using, the quadrature points are also the points used to sample the solutions of the SPDE

Sampling and simple averaging quadrature rules

- We consider sampling + simple averaging-based quadrature rules that are based on
  - determining a set of quadrature points $\{y_q\}_{q=1}^{Q}$
  - approximating integrals of a function $G(y)$ by an equal weight rule

$$\int_\Gamma G(\vec{y})\rho(y)\,dy \approx \frac{1}{Q}\sum_{q=1}^{Q} G(\vec{y}_q)$$

if one samples the points according to the PDF $\rho(\vec{y})$

or by

$$\int_\Gamma G(\vec{y})\rho(y)\,dy \approx \frac{1}{Q}\sum_{q=1}^{Q} \rho(y_q)G(y_q)$$

if one samples the points uniformly

- The second approach seems simpler, but is wasteful
  - the density of points is the same in regions where $\rho(\cdot)$ is small as where it is large
    - unfortunately, many sampling methods can only be used to sample uniformly or have difficulty, i.e., they are much less efficient, when sampling nonuniformly

- Note that the weights do not depend on the position of the points $\{\vec{y}_q\}_{q=1}^{Q}$ or on other geometric quantities

Monte Carlo sampling

- As has already been said, the simplest quadrature rule is based on Monte Carlo, i.e., random, sampling of the hypercube
  - random sampling could be done uniformly in the hypercube
    - in which case $w_q = \dfrac{\rho(\vec{y}_q)}{Q}$
  - random sampling could instead be done according to the density function $\rho(\vec{y})$ by, e.g., a rejection method
    - in which case $w_q = \dfrac{1}{Q}$
- Monte Carlo integration has one very great virtue (other than its simplicity)
  - its convergence behavior is independent of the dimension N, i.e., of the number of parameters

- Unfortunately, it also has one great fault
  - its convergence behavior is slow Error $= O\left(\frac{\sigma}{\sqrt{Q}}\right)$
- The slow convergence of Monte Carlo integration has motivated the huge amount of effort devoted to improving or replacing Monte Carlo sampling as an integration rule
  - it has also motivated the development of stochastic Galerkin methods

"Improved" sampling + simple averaging-based quadrature rules

- There have been many sampling + simple averaging-based quadrature rules proposed as replacements for Monte Carlo quadrature, including

  variance reduction Monte Carlo methods

  quasi-Monte Carlo methods (Halton, Sobol, Faure, Hammersley, ...)

  stratified sampling

  Latin hypercube sampling and its many "improved" versions

  orthogonal arrays

  lattice rules

  importance sampling

  etc.

- In general, these "improved" rules have, in theory, improved rates of convergence, at least for not too large N

  – the best theoretical result is of the type

$$\text{Error} = O\left(\frac{(\ln Q)^N}{Q}\right) \qquad \Longleftarrow \text{ note the dependence on } N$$

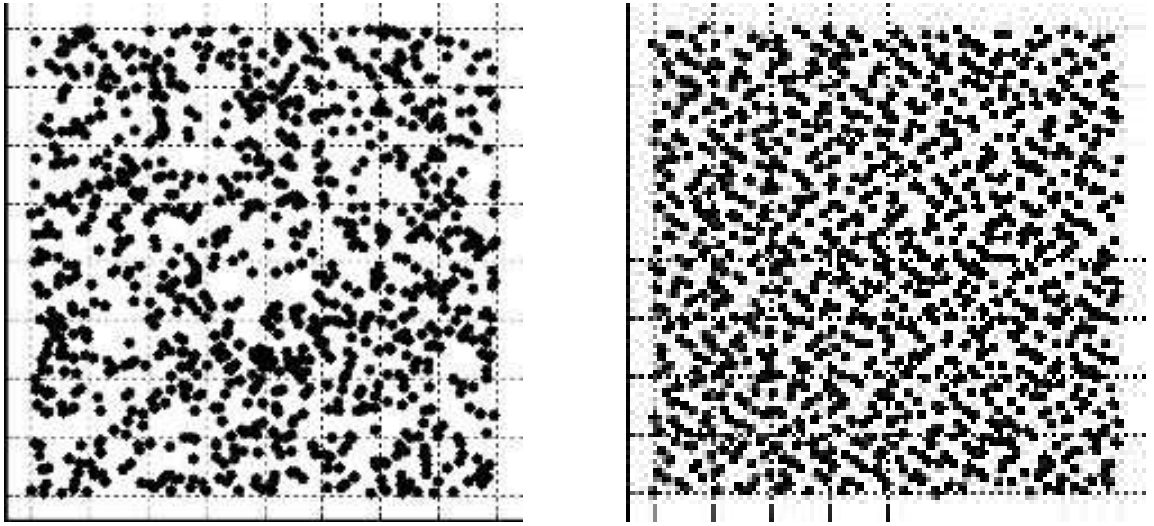
  – this is often a pessimistic estimate

  – for large N, the $(\ln Q)^N$ term dominates

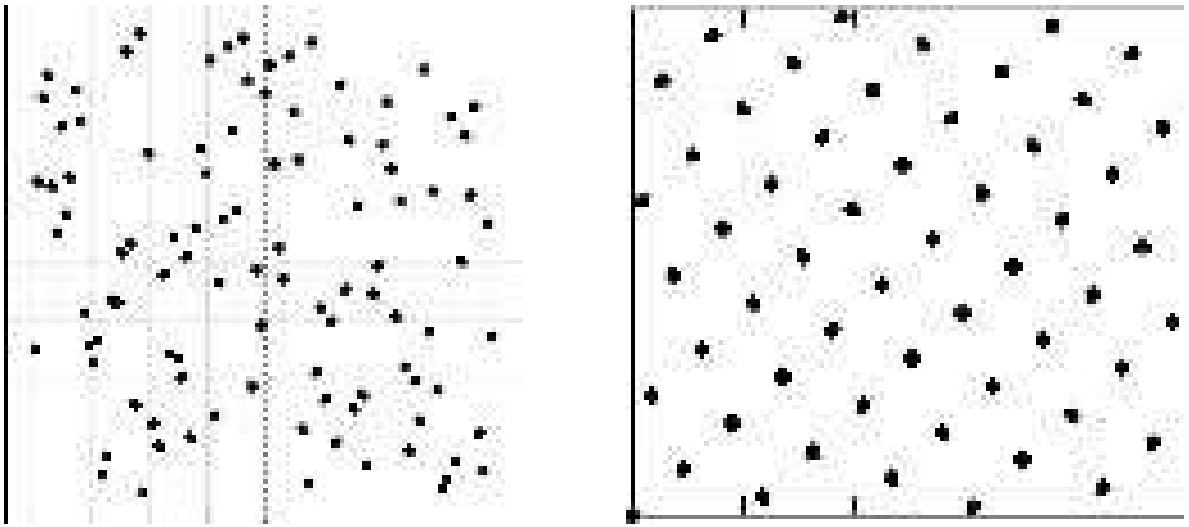
    - the curse of dimensionality is still with us

  – also, in many cases, biasing problems exist, especially for a large number of sample points

- However, if one is careful when using them, the "improved" sampling and averaging methods often can indeed improve on Monte-Carlo sampling



Monte Carlo and quasi-Monte Carlo point sets



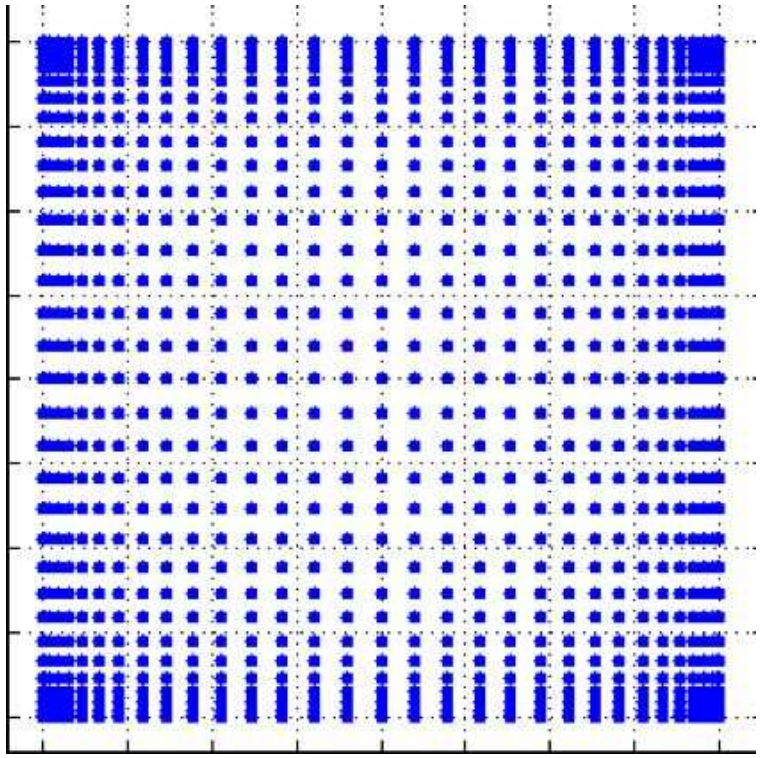
Latin hypercube and lattice rule point sets

Tensor products of standard 1-D quadrature rules

- One is familiar with many quadrature rules in 1D
- On the hypercube, one can easily define multiple integration rules as tensor products of 1D rules
- As we have already seen, tensor products really suffer from the curse of dimensionality
- Tensor product rules integrate tensor products of polynomials exactly
- Just as was the case for interpolation and approximation, one can get the same rate of convergence using quadrature rules that integrate complete polynomials exactly
- The same table of numbers used before applies here

Quadrature rules in hypercubes

$N =$ no. random parameters	number of quadrature points in each direction	$Q =$ number of quadrature points	
		using complete polynomial rule	using a tensor product rule
3	4	20	64
	6	56	216
5	4	56	1,024
	6	252	7,776
10	4	286	1,048,576
	6	3,003	60,046,176
20	4	1,771	$> 1\times10^{12}$
	6	53,130	$> 3\times10^{15}$
100	4	176,851	$> 1\times10^{60}$
	6	96,560,646	$> 6\times10^{77}$



A tensor product set of quadrature points in 2D

- On the other hand, tensor product rules are easy to define
  - the quadrature points are tensor products of the quadrature points of the 1D rules
  - the quadrature weights are products of the weights of the 1D rules

- High-dimensional rules based on complete polynomials are not so easy to define
  - determining a good set of quadrature points and the corresponding quadrature weights is difficult
  - these difficulties further motivated interest in SGM methods

- But now, there is available an intermediate means of defining quadrature rules
  - the number of points is much less that that for tensor product rules, but is somewhat greater than that for complete polynomial rules
  - these rules are constructed through judicious sparsifications of tensor product rules
  - the are known as Smolyak or sparse grid quadrature rules

SPARSE (SMOLYAK) QUADRATURE RULE-BASED STOCHASTIC SAMPLING METHODS

- Let I be a positive integer and for each $i = 1, \ldots, I$,
  let m_i denote a positive integer

- For each $i = 1, \ldots, I$, let $\Theta^{(i)} = \{y_1^{(i)}, \ldots, y_{m_i}^{(i)}\}$
  denote a set of points in $[-1, 1]$

  – note that for convenience, we will be looking at the hypercube $[-1, 1]^N$

- Let $N > 1$ denote the number of parameters

- Let $p = (p_1, p_2, \ldots, p_N)$ denote a multi-index,

  – in this case, an N-vector whose components are positive integers

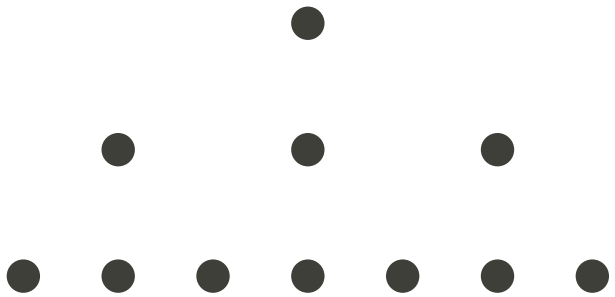
  and let $|p| = \sum_{n=1}^{N} p_n$

- Let M denote a positive integer
- Let $\mathcal{I}(M,N) = \{p \ : \ M+1 \leq |p| \leq N+M\}$
- Then,

$$\mathcal{S}(M,N) = \bigcup_{p \in \mathcal{I}(M,N)} \Theta^{(p_1)} \otimes \Theta^{(p_2)} \otimes \cdots \otimes \Theta^{(p_N)}$$

defines a sparse grid

- Example
  - let $I = 3$, $m_1 = 1$, $m_2 = 3$, and $m_3 = 7$
  - let $\Theta^{(i)}$, $i = \ldots, I = 3$ be given by the three one-dimensional nested point sets



  - let $N = 2$ and $M = 2$ so that $\mathcal{I}(2,2) = \{p \ : \ 3 \leq |p| \leq 4\}$
  - $\mathcal{I}(2,2)$ then contains the combinations

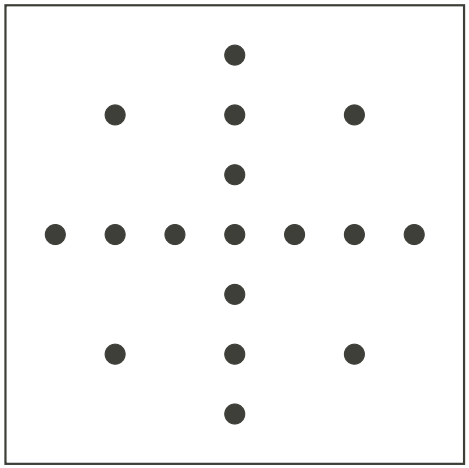
$$(p_1, p_2) = (1,1),\ (1,2),\ (2,1),\ (3,1),\ (1,3),\ (2,2)$$

    but not the combinations

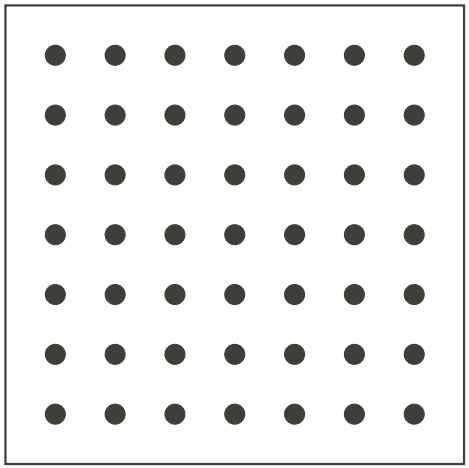
$$(p_1, p_2) = (2,3),\ (3,2),\ (3,3)$$

    - for nested point sets, it is enough to include the combinations for which $|p| = N + M$, i.e., $(3,1),\ (1,3),\ (2,2)$ in the example

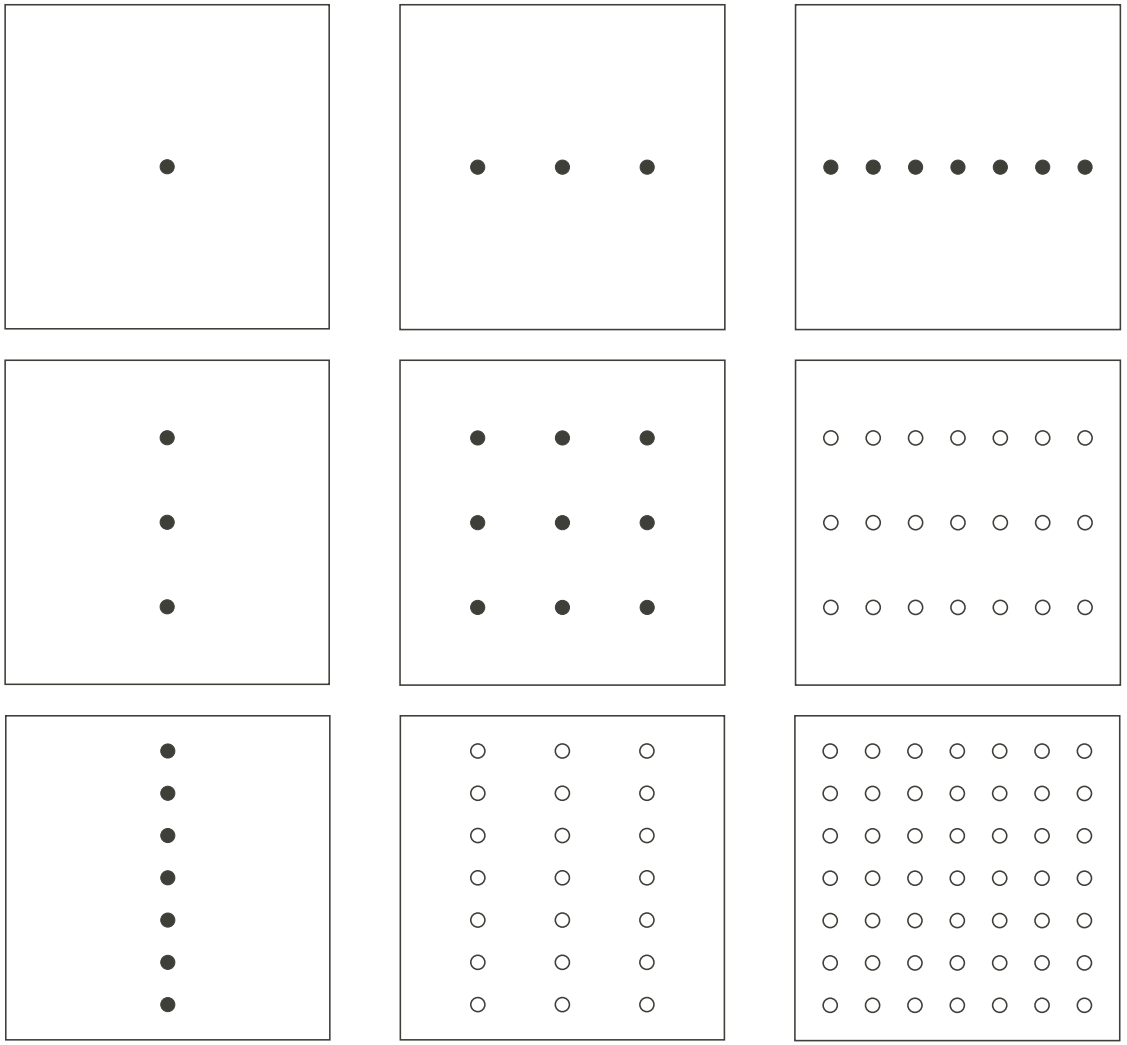
– then, $\mathcal{S}(2,2)$ is given by



– this should be contrasted with the full tensor-product point set



- the following diagram shows how the sparse grid comes about



- point sets included in $\mathcal{S}(2,2)$ ∘ point sets not included in $\mathcal{S}(2,2)$

- What Smolyak showed is that
  - if one chooses the underlying one-dimensional grids to be the quadrature points for some integration rule

  then
  - the accuracy of the full tensor product point set can be preserved with point sets with much fewer points

- Along the way, Smolyak also showed how to systematically compute the weights of the resulting sparse quadrature rule

- The use of Smolyak grids in the SPDE setting has been rigorously analyzed for some simple linear and nonlinear elliptic PDEs

- Some choices of one-dimensional quadrature rules upon which the Smolyak grids can be constructed

  – Newton-Cotes: nested equidistant abscissas by taking $m_1 = 1$ and $m_i = 2^{i-1} + 1$ for $i > 1$

    - maximum degree of exactness is $m_I - 1$

    - can have (highly) negative weights causing numerical inaccuracies

  – Clenshaw-Curtis: nested (same growth as above) Chebyshev points

    - maximum degree of exactness is $m_I - 1$

    - nested grids keep the number of points down

  – Gauss: non-nested abscissas

    - maximum degree of exactness is $2m_I - 1$

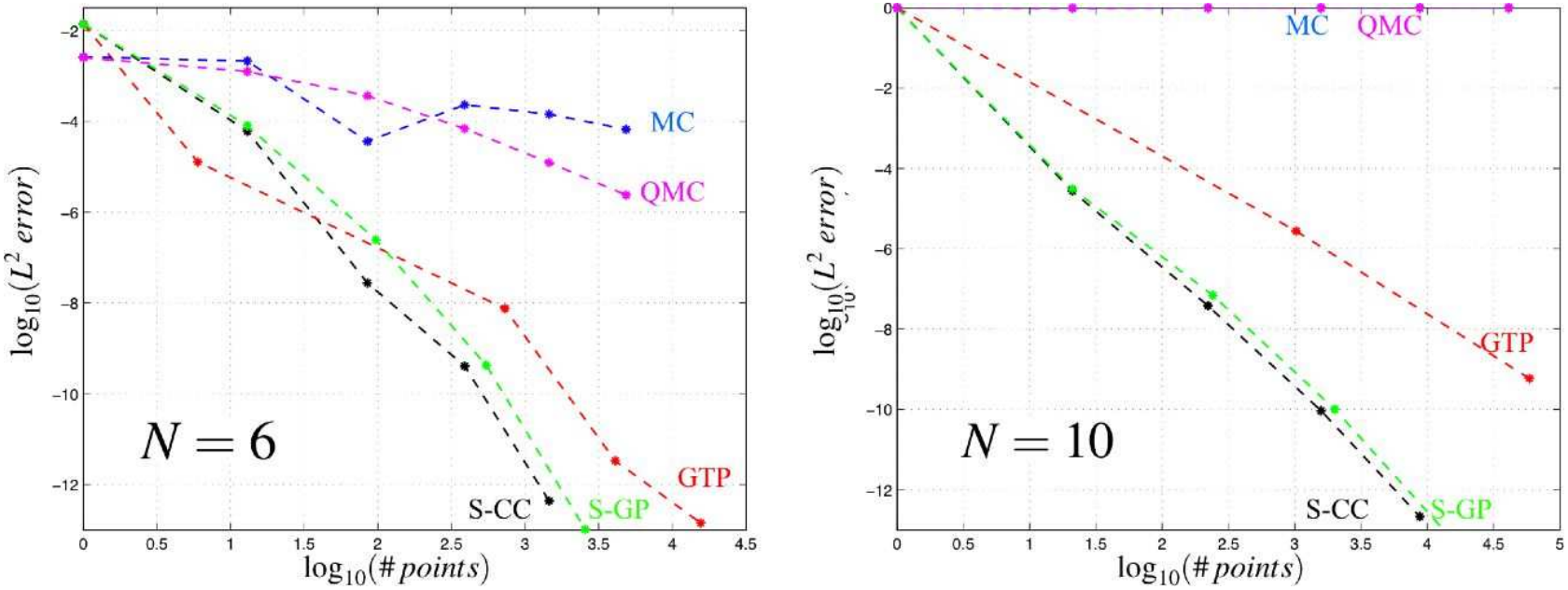
  – Gauss-Patterson: seems to have good promise

Results that follow are from papers of Nobile, Tempone, and Webster

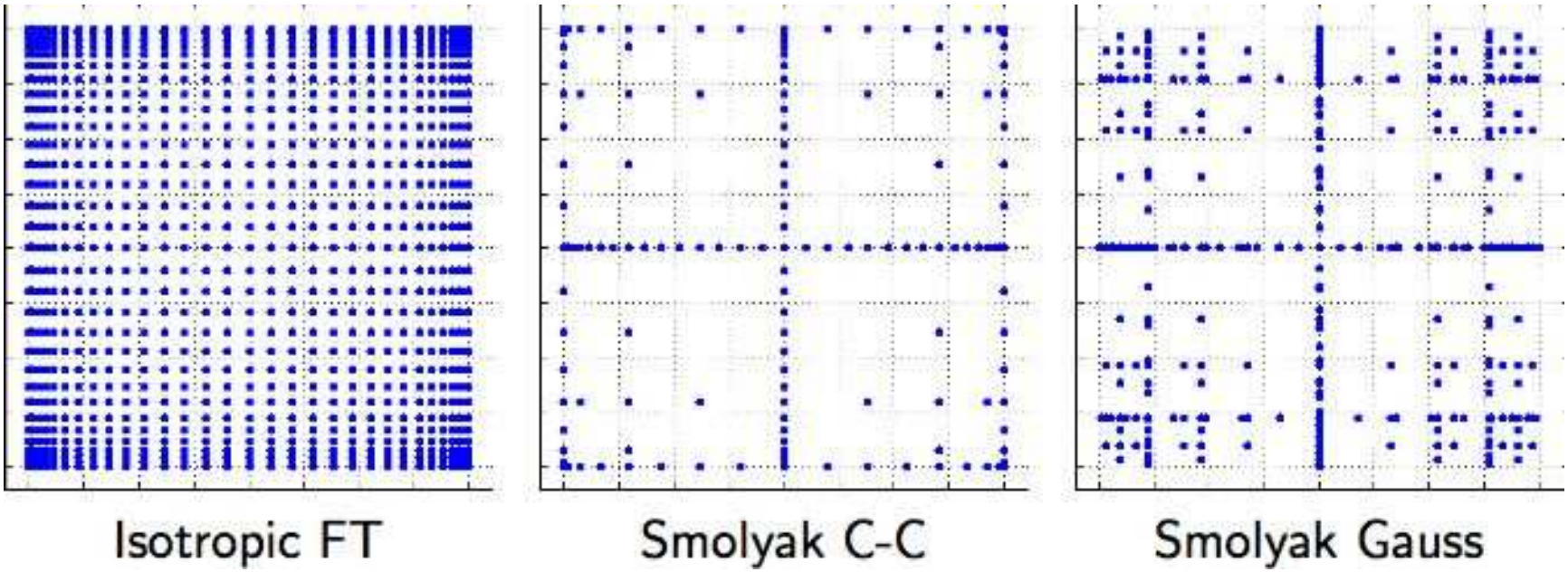
- For the integral

$$\int_{\mathbb{R}^N} \exp\left(-\sum_{n=1}^{N} a_n^2 (y_n - b_n)^2\right) d\vec{y}$$

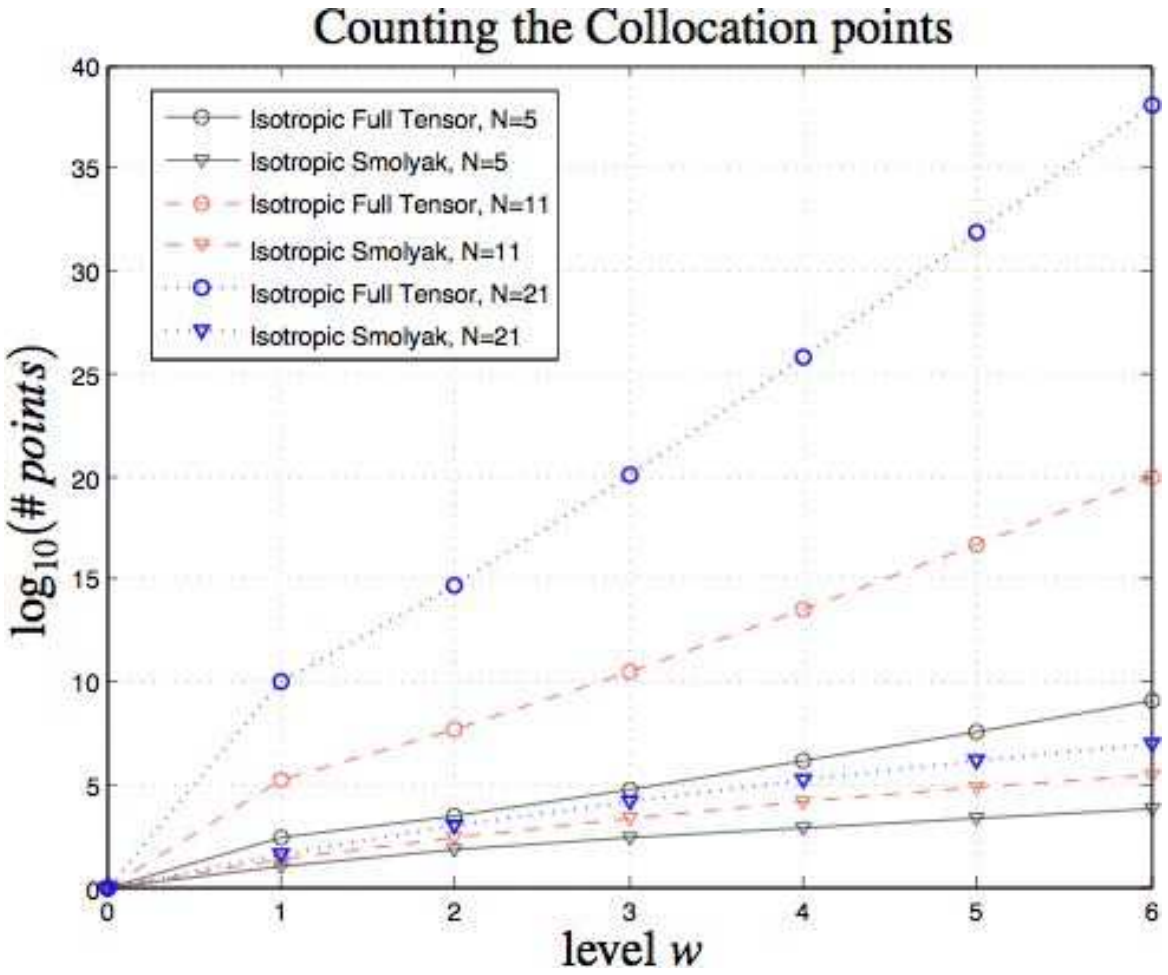
where a_n and b_n are randomly sampled uniformly in $(0, 1)$, we have the following errors for different quadrature rules



Comparisons of errors vs. number of quadrature points for different integration rules

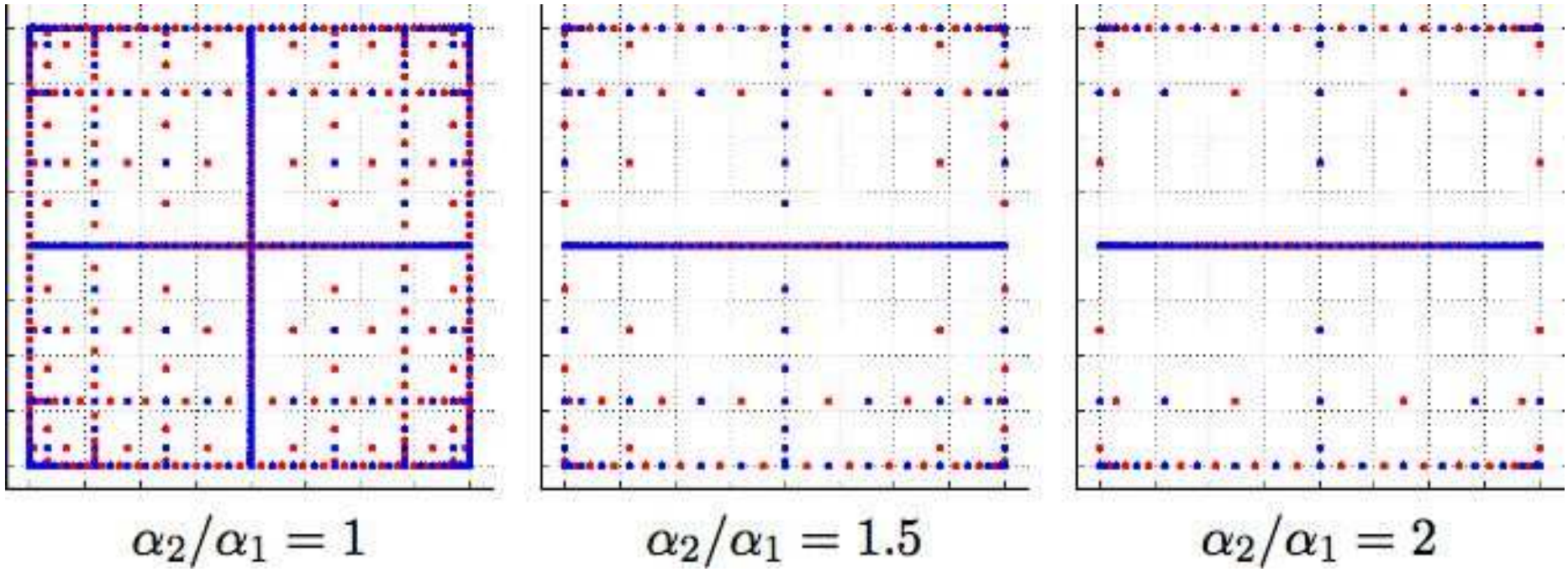


For $N = 2$ and $M = 5$: comparison of full tensor product grids with two Smolyak grids

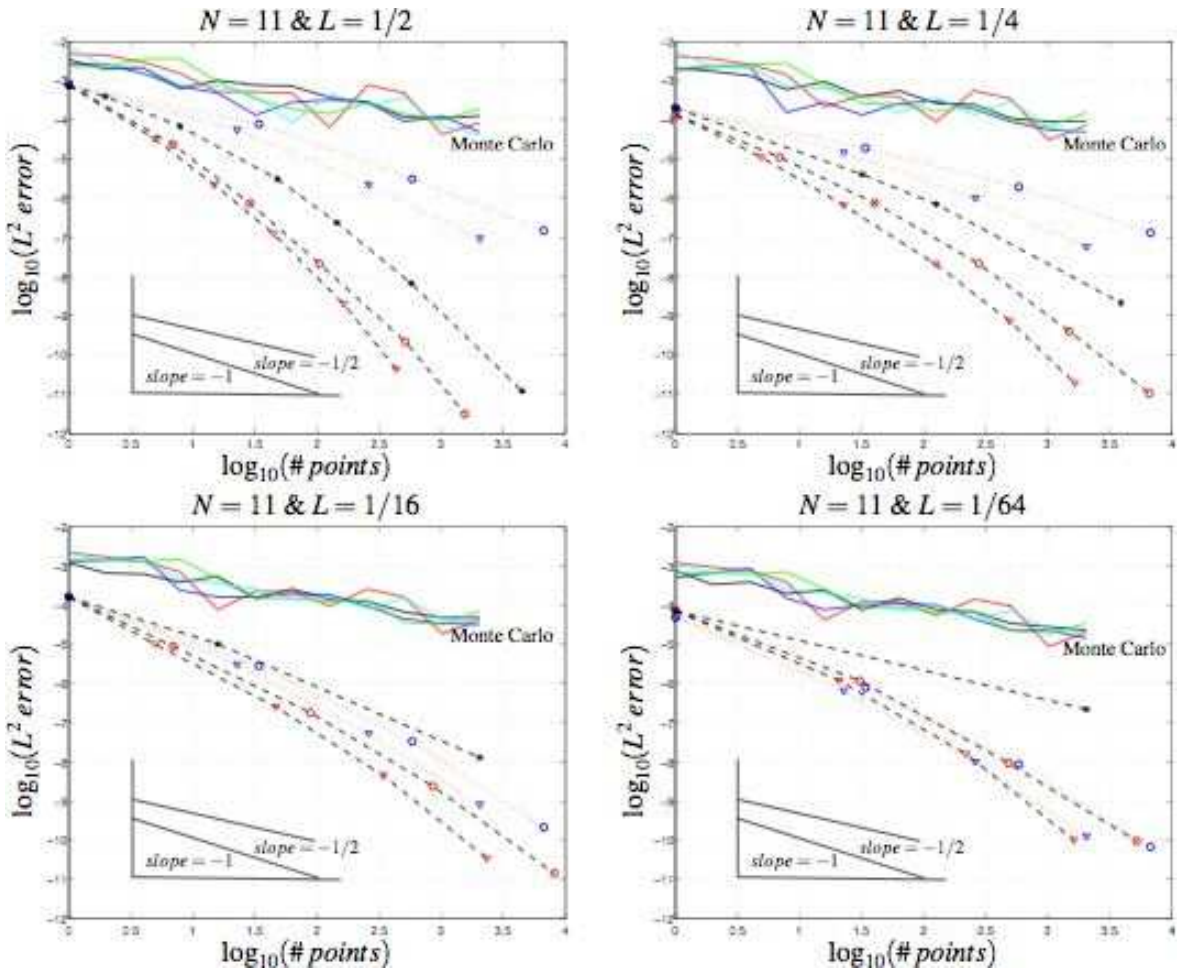


For $N = 5$, 11, and 21: comparison of full tensor product grids with Clenshaw-Curtis-Smolyak grids for different levels, i.e., for different maximum number of points in each direction

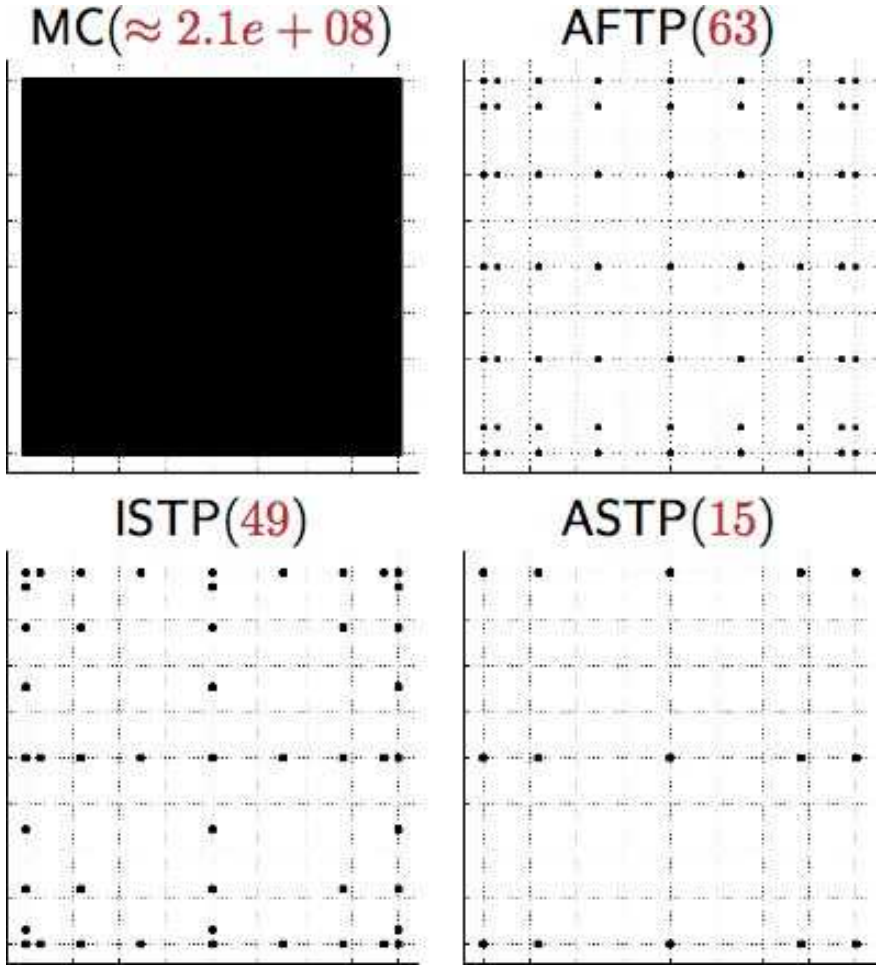
- There is more good news about Smolyak grids
- Recently, anisotropic Smolyak grids have been developed to take advantage of anisotropies in the relative importance of random parameters
- For example, in the Karhunen-Loéve expansion for the colored noise case, the random variables $y_1, y_2, \ldots$ are increasingly less influential
- Adaptive strategies have been developed to determine how to take advantage of such anisotropies



Anisotropic Clenshaw-Curtis sparse grids for different levels of anisotropy; on the left is the isotropic case; the anisotropic grids will yield the same accuracy as the isotropic one, provided the integrand possesess the necessary anisotropy



L^2 errors in the expected values of the solution of an SPDE using different sampling strategies; Monte Carlo is always worst, anisotropic Smolyak best, with Clenshaw-Curtis being better than Gauss; L is a correlation length for the colored noise



Number of points needed to reduce to reduce the L^2 errors in the expected values of the solution of an SPDE by a factor of 10^4

- This shows the effectiveness of using stochastic sampling methods along with modern sparse grid techniques

LOCAL POLYNOMIAL APPROXIMATING SPACES IN STOCHASTIC GALERKIN METHODS

PIECEWISE POLYNOMIAL APPROXIMATING SPACES FOR PARAMETER SPACE DISCRETIZATION

- Emulating finite element spatial discretization methods, one is led to locally-supported piecewise polynomial spaces for approximating functions of the random parameters

- One starts by "triangulating" Γ, the set of all possible values for the random parameters $\{y_1, \ldots, y_N\}$

  – of course, unless one wants to get fancy, i.e.,

    - use infinite elements or other methods for treating unbounded domains

  we have to assume that Γ is bounded

  – thus, we consider problems for which the Γ_n, $n = 1, \ldots, N$, themselves are bounded

    - e.g., we cannot consider y_1 to be a Gaussian random parameter since, in this case, $\Gamma_1 = (-\infty, \infty)$

    - of course, we can considered truncated Gaussian parameters

- One then chooses Z_K to be a space of piecewise polynomial functions of degree less than of equal to M, defined with respect to the triangulation
  - since $Z_K \subset L^q_\rho(\Gamma)$, one can choose $M = 0$, i.e., piecewise constant functions
  - however, one can choose higher degree piecewise polynomials as well
  - one is free to choose discontinuous finite element spaces

- Unfortunately, the number of parameters N cannot be large
  - even for a subdivision with two elements in each direction, N cannot be big, e.g., $K = 2^N$ becomes prohibitively large very quickly

- Also, triangulating in high dimensions is not an easy task
  - unless N is small, one can in practice only consider the case of Γ being rectangular domain in $\mathbb{R}^N$ that is "triangulated" into smaller rectangular domains

- One can choose a standard "finite element"-type basis set
  - $\{\psi_k(\vec{y})\}_{k=1}^K$ consists of compactly supported piecewise polynomials
  - if Z_K is a discontinuous (with respect to the triangulation of Γ) finite element space, then each basis function can be chosen to have support over only a single element
  - if Z_K is a continuous (with respect to the triangulation of Γ) finite element space, then each basis function can be chosen to have support over a small patch of elements

- There is a really big difference between using discontinuous and continuous finite element-type spaces to discretize in parameter space

- First, consider an example of a continuous finite element-type space
  - Γ is a hypercube in N-dimensions (N = number of random parameters)
  - Γ is subdivided into $N_{hypercubes}$ smaller hypercubes
  - Z_K consists of tensor products of continuous piecewise polynomials of degree less that or equal to $M \geq 1$ in each parameter direction
  - then, the number of probabilistic degrees of freedom is given by

$$K = \left(M N_{hypercubes}^{1/N} + 1 \right)^N$$

  - as always, the discrete problem involves JK degrees of freedom $c_{j,k}$

- If we look at the $JK \times JK$ coefficient matrix for the discrete system (emanating from a linear Poisson problem)

$$\int_{\Gamma} \int_{\mathcal{D}} a(\mathbf{x}; \vec{y}) \nabla \phi_j(\mathbf{x}) \cdot \nabla \phi_{j'}(\mathbf{x}) \, \psi_k(\vec{y}) \psi_{k'}(\vec{y}) \rho(\vec{y}) \, d\mathbf{x} d\vec{y}$$

  we see that it is sparse with respect to both the spatial and probabilistic indices

  - if the support of $\phi_j(\mathbf{x})$ and $\phi_{j'}(\mathbf{x})$ do not overlap, then the corresponding matrix entry vanishes for all k and k'
  - if the support of $\psi_k(\mathbf{x})$ and $\psi_{k'}(\mathbf{x})$ do not overlap, then the corresponding matrix entry vanishes for all j and j'
  - this sparsity can be taken advantage of when one solves the system, especially if one uses an iterative method
  - however, we still have a coupled (albeit sparse) $JK \times JK$ system to solve

- Now, consider an example of using discontinuous finite element-type spaces to discretize in parameter space

  – Γ is a hypercube in N-dimensions (N = number of random parameters)

  – Γ is subdivided into $N_{hypercubes}$ smaller hypercubes

  – in each element, Z_K consists of complete polynomials of degree less that or equal to $M \geq 0$

    - no continuity is required across element boundaries

  – then, the number of probabilistic degrees of freedom is given by

$$K = N_{hypercubes} \left(\frac{(N+M)!}{N!M!} \right)$$

    which can be larger than that obtained using continuous finite element-type spaces

  – as always, the discrete problem involves JK degrees of freedom $c_{j,k}$

Piecewise polynomial approximation in parameter space

$N =$ no. random parameters	$M =$ maximal degree of polynomials	$N_{hypercubes}^{1/N} =$ no. of intervals in each direction	$K =$ no. of probabilistic degrees of freedom	
			continuous tensor product basis	discontinuous basis
3	0	5	–	125
		10	–	1,000
	1	5	216	500
		10	1,331	4,000
	2	5	1,331	1,250
		10	9,261	10,000
5	0	5	–	3,125
		10	–	100,000
	1	5	7,776	18,750
		10	161,051	600,000
	2	5	161,051	65,625
		10	4,084,101	2,100,000

- But, let's examine the $JK \times JK$ coefficient matrix for the discrete system in the discontinuous finite element case

$$\int_{\Gamma} \int_{\mathcal{D}} a(\mathbf{x}; \vec{y}) \nabla \phi_j(\mathbf{x}) \cdot \nabla \phi_{j'}(\mathbf{x}) \, \psi_k(\vec{y}) \psi_{k'}(\vec{y}) \rho(\vec{y}) \, d\mathbf{x} d\vec{y}$$

  - again, we have the usual sparsity with respect to both the spatial indices
  - but now, since the support of the probabilistic basis functions $\{\psi_k(\vec{y})\}_{k=1}^{K}$ is restricted to a single element in parameter space, we have that
    - with respect to parameter space, the matrix is block diagonal
    - there is a complete uncoupling of the probabilistic degrees of freedom

- Let $\Gamma_{hypercube}$ denote one of the $N_{hypercubes}$ elements in the subdivision of Γ into smaller hypercubes

- Let $K_{hypercube}$ denote the probabilistic degrees of freedom in each element $\Gamma_{hypercube}$, i.e.,

$$K_{hypercube} = \frac{(N+M)!}{N!M!} = \frac{K}{N_{hypercubes}}$$

- For each of the $N_{hypercubes}$ elements $\Gamma_{hypercube}$, let

$$I_{hypercube} = \left\{ k \in \{1, \ldots, K\} \mid \text{supp}\big(\psi_k(\vec{y})\big) \subset \Gamma_{hypercube} \right\}$$

  – note that the cardinality of the index set $I_{hypercube}$ is $K_{hypercube}$

- Then, the coupled $JK \times JK$ system for the degrees of freedom $c_{j,k}$ uncouples into $N_{hypercubes}$ systems, each of size $JK_{hypercube} \times JK_{hypercube}$

$$\int_{\mathcal{D}} \int_{\Gamma} \rho(\vec{y}) S\Big(\sum_{j=1}^{J} \sum_{k=1}^{K} c_{jk} \phi_j(\mathbf{x}) \psi_k(\vec{y}), \vec{y} \Big) T\Big(\phi_{j'}(\mathbf{x}) \Big) \psi_{k'}(\vec{y}) \, d\mathbf{x} d\vec{y}$$

$$= \int_{\mathcal{D}} \int_{\Gamma} \rho(\vec{y}) \phi_{j'}(\mathbf{x}) \psi_{k'}(\vec{y}) f(\vec{y}) \, d\mathbf{x} d\vec{y}$$

for $j' \in \{1, \ldots, J\}$ and $k' \in \{1, \ldots, K\}$

$$\sum_{j=1}^{J} \sum_{k \in I_{hypercube}} c_{j,k} \int_{\Gamma_{hypercube}} \int_{\mathcal{D}} a \nabla \phi_j(\mathbf{x}) \cdot \nabla \phi_{j'}(\mathbf{x}) \, \psi_k(\vec{y}) \psi_{k'}(\vec{y}) \rho(\vec{y}) \, d\mathbf{x} d\vec{y}$$

$$= \int_{\Gamma_{hypercube}} \int_{\mathcal{D}} f \phi_{j'}(\mathbf{x}) \psi_{k'}(\vec{y}) \rho(\vec{y}) \, d\mathbf{x} d\vec{y}$$

for all $j' = 1, \ldots, J$ and $k' \in I_{hypercube}$

- The moral of the story is that, in practice, once pretty much has to settle for piecewise constant approximations in parameter space

- Even for this case, N cannot be too large

PIECEWISE CONSTANT APPROXIMATING SPACES

- Let $\cup_{k=1}^{K} \Gamma_k$ denote a subdivision of Γ into disjoint, non-overlapping subsets

  – we have that

$$\cup_{k=1}^{K} \overline{\Gamma}_k = \overline{\Gamma} \qquad \text{and} \qquad \Gamma_k \cap \Gamma_{k'} = \emptyset \quad \text{if } k \neq k'$$

- Let

$$\psi_k(\vec{y}) = \begin{cases} 1 & \text{if } \vec{y} \in \Gamma_k \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } k \in \{k, \ldots, K\}$$

  and let

$$Z_K = \text{span}\, \{\psi_k\}_{k=1}^{K}$$

  – thus, Z_K is the space of piecewise constant functions with respect to the partition $\cup_{k=1}^{K} \Gamma_k$ of Γ

- Clearly, $Z_K \subset L^p_\rho(\Gamma)$ so that it can be used as an approximating space for discretizing parameter dependences of solution of an SPDE

- Recall that, after the invocation of the piecewise constant basis functions and of a parameter-space quadrature rule, the stochastic Galerkin method has the form

$$\sum_{r=1}^{R} w_r \rho(\vec{y}_r) \psi_{k'}(\vec{y}_r) \int_{\mathcal{D}} S\Big(\sum_{j=1}^{J} \sum_{k=1}^{K} c_{jk} \phi_j(\mathbf{x}) \psi_k(\vec{y}_r), \vec{y}_r \Big) T\Big(\phi_{j'}(\mathbf{x}) \Big)\, d\mathbf{x}$$

$$= \sum_{r=1}^{R} w_r \rho(\vec{y}_r) \psi_{k'}(\vec{y}_r) \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\vec{y}_r)\, d\mathbf{x}$$

for $j' \in \{1, \ldots, J\}$ and $k' \in \{1, \ldots, K\}$

where $\{w_r, \vec{y}_r\}_{r=1}^R$ denotes the quadrature rule used to approximate integrals over parameter space Γ

- Suppose we choose the quadrature rule so that

$$R = K \qquad \text{and} \qquad \vec{y}_r \in \Gamma_r \quad \text{for } r \in \{1, \ldots, R = K\}$$

  – thus,

    - each quadrature point $\vec{y}_r$ belongs to one of the subsets Γ_k

  and

    - each subset contains one and only one of the quadrature points

  – Clearly, we then have that

$$\psi_k(\vec{y}_r) = \delta_{kr} \qquad \text{for all } k, r \in \{1, \ldots, K = R\}$$

- Then, the discretized stochastic Galerkin system reduces to

$$\int_{\mathcal{D}} S\Big(u_r(\mathbf{x}), \vec{y}_r\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\vec{y}_r)\, d\mathbf{x}$$

$$\text{for } j' \in \{1, \ldots, J\} \quad \text{and} \quad r \in \{1, \ldots, R = K\}$$

where $u_r(\mathbf{x}) = \sum_{j=1}^{J} c_{jr}\phi_j(\mathbf{x})$

  - thus, we have total uncoupling of the spatial and parameter problems
  - we solve a sequence of $R = K$ problems of size J to determine $\{u_r(\mathbf{x})\}_{r=1}^{R}$
  - then, the stochastic Galerkin-piecewise constant approximation of the solution of the SPDE is simply given by

$$u(\mathbf{x}; \vec{y}) = u_r(\mathbf{x}) \qquad \text{for } \vec{y} \in \Gamma_r$$

- Note that to determine the $u_r(\mathbf{x})$ one does not have to explicitly know the weights w_r or the subregions Γ_k

  – one need only know the point set $\{\vec{y}_k\}_{k=1}^K$

- Note also that there is no restrictions on the point set $\{\vec{y}_k\}_{k=1}^K$

  – one can, in fact, use any of the point sets we have encountered in discussing stochastic sampling or stochastic collocation or stochastic Galerkin methods

- Clearly,

any stochastic sampling method can be viewed as a stochastic Galerkin method

Approximations of quantities of interest

- It is natural to use the same quadrature rule

  - to approximate a quantity of interest

  as was used to

  - approximate the integrals in discretized SPDE,

  i.e., we choose

$$K = R = Q$$

$$\{\vec{y}_k\}_{k=1}^{K} = \{\vec{y}_r\}_{r=1}^{R} = \{\vec{y}_q\}_{q=1}^{Q} \qquad \text{and} \qquad \{w_r\}_{r=1}^{R} = \{w_q\}_{q=1}^{Q}$$

- We then have that

$$\psi_r(\vec{y}_q) = \delta_{rq} \qquad \text{for all } r, q \in \{1, \ldots, K = R = Q\}$$

- Using this in the expression for the approximation of a quantity of interest results in

$$\int_\Gamma G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_{SC}(\mathbf{x})\Big)$$

$$= \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(\sum_{r=1}^{R} u_r(\mathbf{x})\psi_r(\vec{y}_q)\Big) = \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_q(\mathbf{x})\Big)$$

i.e.,

$$\int_\Gamma G\Big(u(\mathbf{x};\vec{y})\Big)\rho(\vec{y})\,d\vec{y} \approx \sum_{q=1}^{Q} w_q \rho(\vec{y}_q) G\Big(u_q(\mathbf{x})\Big)$$

where, for $q \in \{1,\ldots,Q = R = K_{LI}\}$, $u_q(\mathbf{x}) = \sum_{j=1}^{J} c_{jq}\phi_j(\mathbf{x})$
is determined from

$$\int_{\mathcal{D}} S\Big(u_q(\mathbf{x}),\vec{y}_q\Big)T\Big(\phi_{j'}(\mathbf{x})\Big)\,d\mathbf{x} = \int_{\mathcal{D}} \phi_{j'}(\mathbf{x}) f(\vec{y}_q)\,d\mathbf{x} \qquad \text{for } j' \in \{1,\ldots,J\}$$

- This all looks very familiar:
  - it looks just the same as when we discussed stochastic collocation methods
  - in fact, there is very little distinction between stochastic sampling and stochastic collocation methods
  - and, as we have seen, all stochastic sampling and stochastic collocation methods can be derived from the stochastic Galerkin framework

ECONOMIES IN POLYNOMIAL CHAOS METHODS FOR LINEAR SPDES

- Suppose that the SPDE is linear in the solution u

- For example, consider the case for which one has, after using a polynomial chaos expansion method, the SPDE[†]

$$\int_{\mathcal{D}} \int_{\Gamma} \rho(\vec{y}) a(\mathbf{x}; \vec{y}) S\Big(\sum_{j=1}^{J} \sum_{k=1}^{K_{PC}} c_{jk} \phi_j(\mathbf{x}) \Psi_k(\vec{y}) \Big) T\Big(\phi_{j'}(\mathbf{x}) \Big) \Psi_{k'}(\vec{y}) \, d\mathbf{x} d\vec{y}$$
$$= \int_{\mathcal{D}} \int_{\Gamma} \rho(\vec{y}) \phi_{j'}(\mathbf{x}) \Psi_{k'}(\vec{y}) f(\mathbf{x}; \vec{y}) \, d\mathbf{x} d\vec{y},$$

where now both $S(\cdot)$ and $T(\cdot)$ are linear

[†]Here, it is useful to follow the explicit dependences of the data functions a and f on the spatial variable $\mathbf{x}$

- Since, $S(\cdot)$ is linear and does not involve derivatives with respect to the components of $\vec{y}$, we have that

$$\sum_{j=1}^{J}\sum_{k=1}^{K_{PC}} c_{jk}\int_{\mathcal{D}} S\Big(\phi_j(\mathbf{x})\Big)T\Big(\phi_{j'}(\mathbf{x})\Big)\int_{\Gamma} a(\mathbf{x};\vec{y})\rho(\vec{y})\Psi_k(\vec{y})\Psi_{k'}(\vec{y})\,d\vec{y}d\mathbf{x}$$
$$= \int_{\mathcal{D}} \phi_{j'}(\mathbf{x})\int_{\Gamma} f(\mathbf{x};\vec{y})\rho(\vec{y})\Psi_{k'}(\vec{y})\,d\vec{y}d\mathbf{x}$$

- In this linear SPDE case, there are two economies possible in the implementation of PC methods

PC-expansions of data functions

- We approximate the data functions a and f in the same way one approximates the solution, i.e., using PC-expansions

  – thus, we assume we have in hand the approximations

$$a(\mathbf{x};\vec{y}) \approx \sum_{k''=1}^{K_{PC}} a_{k''}(\mathbf{x})\Psi_{k''}(\vec{y})$$

  and

$$f(\mathbf{x};\vec{y}) \approx \sum_{k''=1}^{K_{PC}} f_{k''}(\mathbf{x})\Psi_{k''}(\vec{y})$$

– substituting into the PC-discretization of the SPDE results in

$$\sum_{k''=1}^{K_{PC}} \sum_{j=1}^{J} \sum_{k=1}^{K_{PC}} c_{jk} \left(\int_{\mathcal{D}} a_{k''}(\mathbf{x}) S\Big(\phi_j(\mathbf{x})\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x} \right) \left(\int_{\Gamma} \rho(\vec{y}) \Psi_k(\vec{y}) \Psi_{k'}(\vec{y}) \Psi_{k''}(\vec{y})\, d\vec{y} \right)$$

$$= \sum_{k''=1}^{K_{PC}} \left(\int_{\mathcal{D}} f_{k''}(\mathbf{x}) \phi_{j'}(\mathbf{x})\, d\mathbf{x} \right) \left(\int_{\Gamma} \rho(\vec{y}) \Psi_{k'}(\vec{y}) \Psi_{k''}(\vec{y})\, d\vec{y} \right)$$

$$= \int_{\mathcal{D}} f_k(\mathbf{x}) \phi_{j'}(\mathbf{x})\, d\mathbf{x}$$

where the last equality follows from the orthonormality of the PC-basis functions $\{\Psi_k(\vec{y})\}_{k=1}^{K_{PC}}$

– orthogonality also results in some sparsity in the left-hand side that may be taken advantage of when using iterative linear system solution methods

- for example, whenever $k + k' \neq k''$ (and for similar situations involving reversal of indices), the summand on the left-hand side vanishes

- Determining the PC-approximations of the data functions a and f may be costly since one has to determine a different expansion for every spatial quadrature point used in the finite element spatial discretization

  – of course, if the data is independent of $\mathbf{x}$, then only one expansion for each data function is needed

- We again point out that the economies resulting from the use of PC-expansions of the data functions are realizable only for linear SPDEs

KL-expansions of random data fields

- Now, suppose that the data functions a and f are Gaussian correlated random fields

  – then, we may determine the approximate KL-expansions

$$a(\mathbf{x};\vec{y}) \approx \sum_{n=1}^{N} \sqrt{\lambda_n} a_n(\mathbf{x}) y_n$$

  and

$$f(\mathbf{x};\vec{y}) \approx \sum_{n=1}^{N} \sqrt{\sigma_n} f_n(\mathbf{x}) y_n,$$

  – $\{\lambda_n, a_n(\mathbf{x})\}_{n=1}^{\infty}$ and $\{\sigma_n, f_n(\mathbf{x})\}_{n=1}^{\infty}$ are the eigenpairs of the covariance functions for a and f, respectively

  – recall that we have to assume (spherical) Gaussian variables since otherwise $\vec{y}$ is not a set of independent parameters

– substituting into the PC-discretization of the linear SPDE results in

$$\sum_{j=1}^{J}\sum_{k=1}^{K_{PC}} c_{jk} \sum_{n=1}^{N} \sqrt{\lambda_n}\left(\int_{\mathcal{D}} a_n(\mathbf{x}) S\Big(\phi_j(\mathbf{x})\Big) T\Big(\phi_{j'}(\mathbf{x})\Big)\, d\mathbf{x}\right) \left(\int_{\Gamma} y_n \rho(\vec{y}) \Psi_k(\vec{y}) \Psi_{k'}(\vec{y})\, d\vec{y}\right)$$

$$= \sum_{n=1}^{N} \sqrt{\sigma_n}\left(\int_{\mathcal{D}} f_n(\mathbf{x}) \phi_{j'}(\mathbf{x})\, d\mathbf{x}\right)\left(\int_{\Gamma} y_n \rho(\vec{y}) \Psi_{k'}(\vec{y})\, d\vec{y}\right)$$

- Doubly orthogonal polynomials can be constructed[†] such that

$$\int_{\Gamma} \Psi_k(\vec{y}) \Psi_{k'}(\vec{y}) \rho(\vec{y})\, d\vec{y} = 0 \qquad \text{and} \qquad \int_{\Gamma} \vec{y}\, \Psi_k(\vec{y}) \Psi_{k'}(\vec{y}) \rho(\vec{y})\, d\vec{y} = 0$$

whenever $k \neq k'$

[†]The construction involves solving an eigenvalue problem for each polynomial

- As a result, the probabilistic and spatial degrees of freedom uncouple
  - one can solve for the c_{ij}'s by solving K_{PC} deterministic finite element problems of size J instead of the single problem of size JK_{PC}

- We again point out that the economies resulting from the use of KL-expansions of the data random fields are realizable only for linear SPDEs

- Moreover, even for linear SPDEs, they are only possible for Gaussian random fields since it is only in this case that the KL expansions are linear in independent random parameters

- This should be contrasted with stochastic collocation methods and the non-intrusive polynomial chaos methods for which the uncoupling of the parameter and spatial degrees of freedom occurs for general, nonlinear SPDEs
  - for stochastic collocation methods, the uncoupling also occurs for general, non-Gaussian probability distributions

OPTIMAL CONTROL PROBLEMS FOR STOCHASTIC PARTIAL DIFFERENTIAL EQUATIONS

Optimization problems

- The state system

$$-\nabla \cdot \big(\kappa(\omega, \mathbf{x}) \nabla u(\omega, \mathbf{x})\big) = f(\omega, \mathbf{x}) \qquad \text{in } \Omega \times D$$

$$u(\omega, \mathbf{x}) = 0 \qquad \text{on } \Omega \times \partial D$$

  - ω is an elementary event in a probability space Ω
  - $\mathbf{x}$ is a point in the spatial domain D
  - $\kappa(\omega, \mathbf{x})$ and $f(\omega, \mathbf{x})$ are correlated random fields
  - the solution $u(\omega, \mathbf{x})$ is also a random field

- Optimal control problem
  - $\kappa(\omega, \mathbf{x})$ is given
  - $f(\omega, \mathbf{x})$ to be determined
  - given target function $\widehat{u}(\omega, \mathbf{x})$ may be deterministic or may be a random field
  - cost functional ($\mathsf{E}(\cdot)$ denotes the expected value)

$$\mathcal{F}(u, f; \widehat{u}) = \mathsf{E}\Big(\|u(\omega, \cdot) - \widehat{u}(\omega, \cdot)\|^2_{L^2(D)} + \alpha \|f(\omega, \cdot)\|^2_{L^2(D)} \Big)$$

$\Longrightarrow$

find a state u and a control f such that $\mathcal{F}(u, f; \widehat{u})$ is minimized subject to the state system being satisfied

- Parameter identification problem
  - $f(\omega, \mathbf{x})$ is given
  - $\kappa(\omega, \mathbf{x})$ to be determined
  - given target function $\widehat{u}(\omega, \mathbf{x})$ may be deterministic or may be a random field
  - cost functional

$$\mathcal{K}(u, \kappa; \widehat{u}) = \mathsf{E}\Big(\|u(\omega, \cdot) - \widehat{u}(\omega, \cdot)\|^2_{L^2(D)} + \beta \|\nabla \kappa(\omega, \cdot)\|^2_{L^2(D)} \Big)$$

$\Longrightarrow$

find a state u and a coefficient function κ such that $\mathcal{K}(u, \kappa; \widehat{u})$ is minimized subject to the state system being satisfied

Results

- Existence of optimal solutions
- Existence of Lagrange multipliers
- Derivation of optimality system
  - the adjoint or co-state system

$$-\nabla \cdot \big(\kappa(\omega, \mathbf{x}) \nabla \xi(\omega, \mathbf{x})\big) = -\big(u(\omega, \mathbf{x}) - \widehat{u}(\omega, \mathbf{x})\big) \qquad \text{in } \Omega \times D$$

$$\xi(\omega, \mathbf{x}) = 0 \qquad \text{on } \Omega \times \partial D$$

  - optimality condition

$$\mathsf{E}\big(-\beta \Delta \kappa + \nabla u \cdot \nabla \xi \big) = 0$$

- Discretization of noise so that κ, f, $\widehat{u}$, and u depend on a parameter vector $\vec{y}(\omega) = (y_1(\omega), \ldots, y_N(\omega))^T$
  - these parameters may be "knobs" in an experiment
  - alternately, they could result from an approximation, e.g., a truncated Karhunen-Loevy expansion, of a correlated random field

- finite element analyses of stochastic collocation method (in progress)
  - isotropic and anisotropic Smolyak sparse grids are used as collocation points

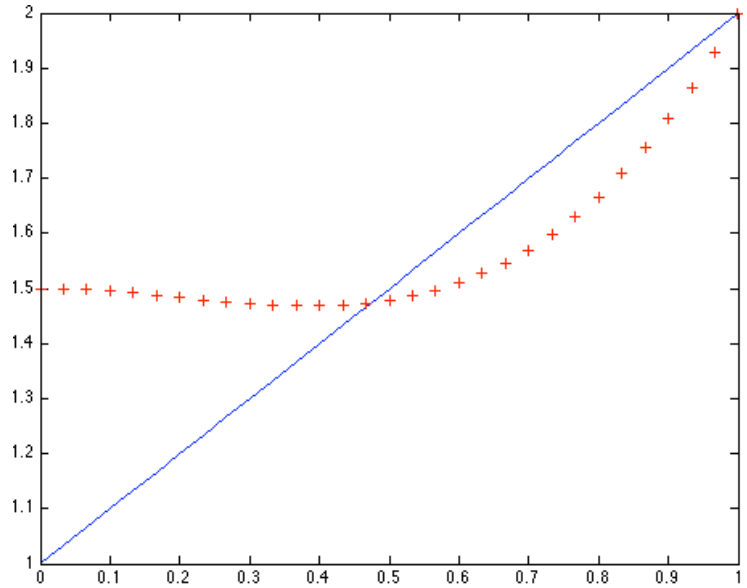
- development of gradient method to effect optimization

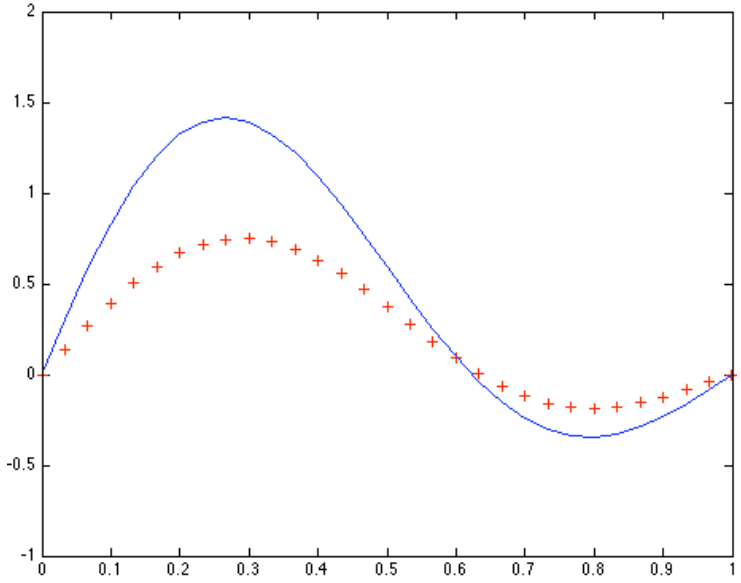
Computational results

- choose target $\widehat{u} = x(1-x^2) + \sum_{i=1}^{N} \sin\left(\frac{n\pi x}{L}\right) y_n(\omega)$
- choose optimal $\kappa = (1+x^3) + \sum_{i=1}^{N} \cos\left(\frac{n\pi x}{L}\right) y_n(\omega)$
- set $f = -\nabla \cdot \left(\kappa \nabla \widehat{u}\right)$
- choose initial $\kappa = 1 + x$
- assume y_i uniform on $[-1, 1]$ with $\mathsf{E}(y_i) = 0$ and $\mathsf{E}(y_i y_j) = \delta_{ij}$

$\Longrightarrow$

given random f and $\widehat{u}$, identify the expectation of both the control $\mathsf{E}(\kappa)$ and the state $\mathsf{E}(u)$ and compare with the exact statistical quantities

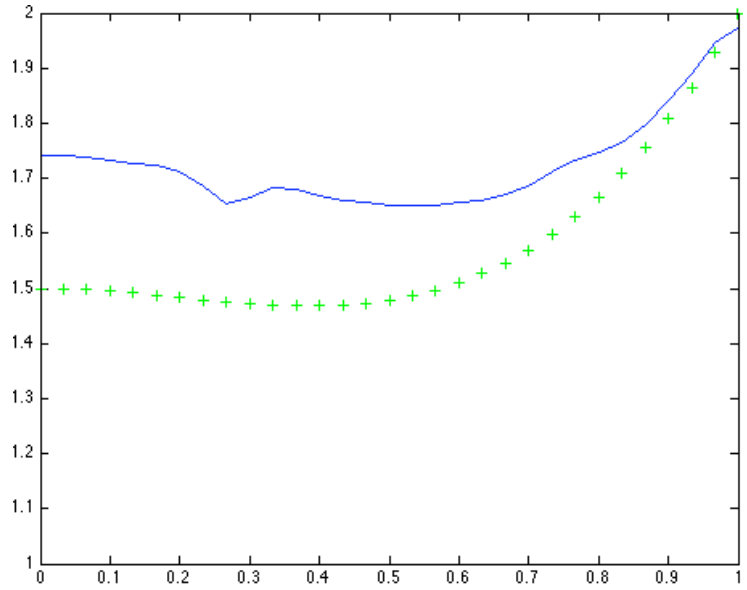


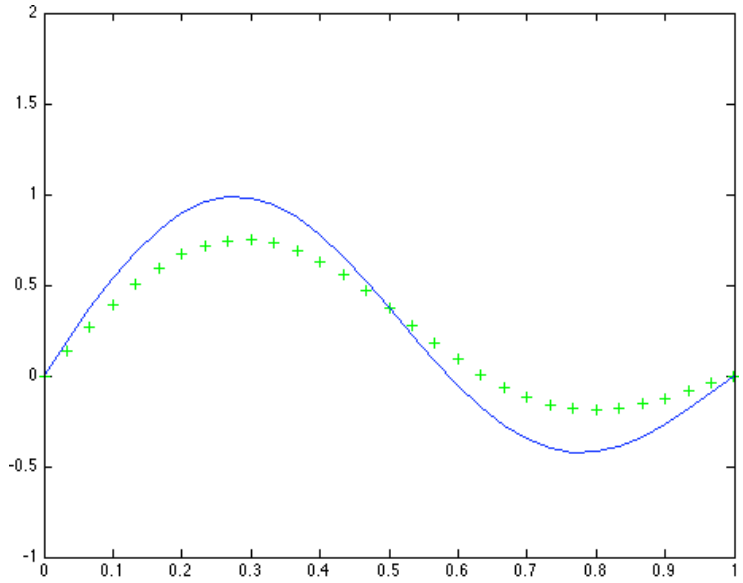


Left: expected value of initial (blue) and target (red) coefficient κ

Right: expected value of initial and target solution u

Number of random variables $= N = 1$



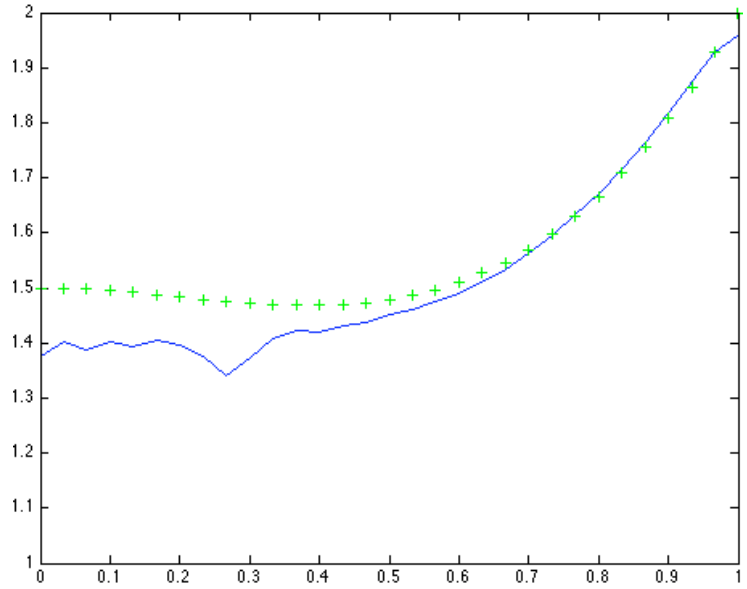


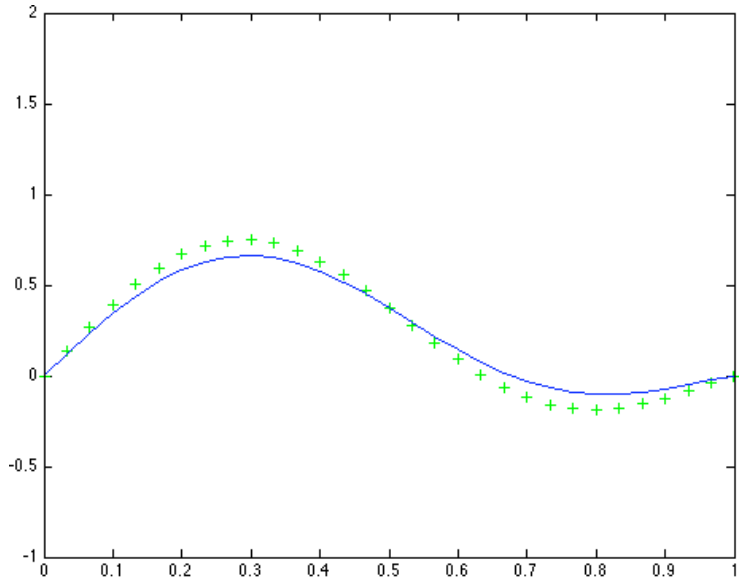
Left: expected value of optimal and target coefficient κ

Right: expected value of optimal and target solution u

Number of random variables $= N = 1$

Number of Monte Carlo samples $= M = 1$



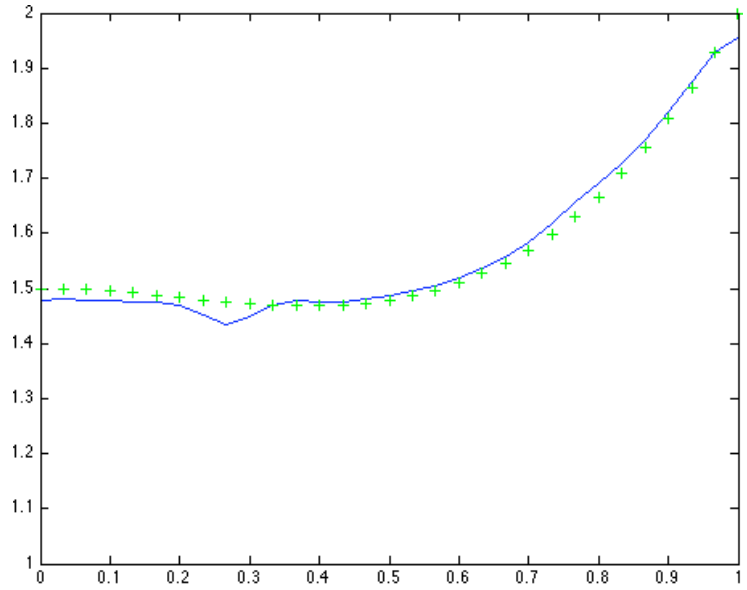


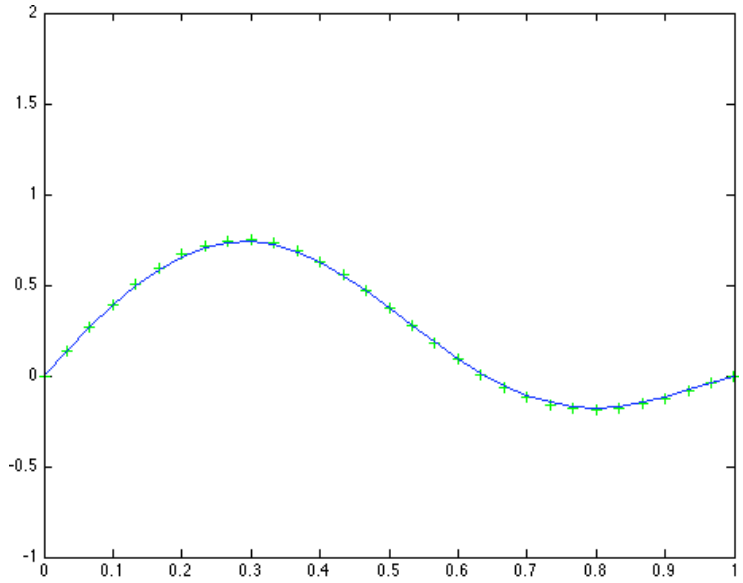
Left: expected value of optimal and target coefficient κ

Right: expected value of optimal and target solution u

Number of random variables $= N = 1$

Number of Monte Carlo samples $= M = 10$



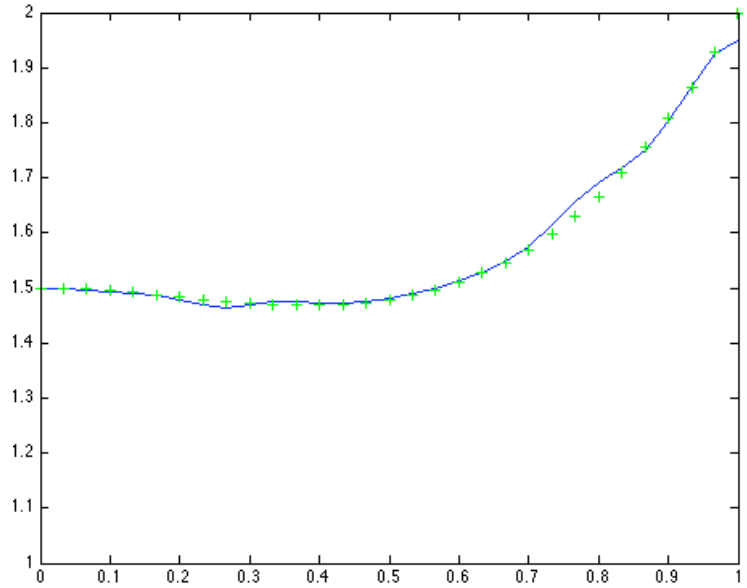


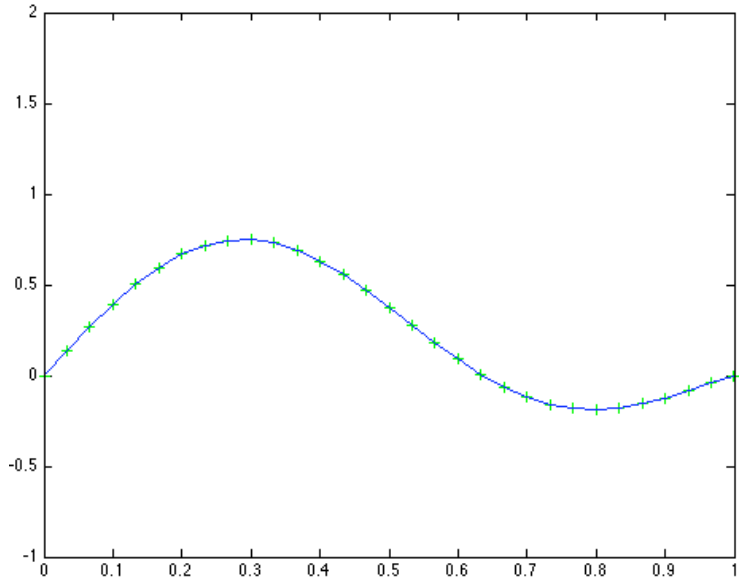
Left: expected value of optimal and target coefficient κ

Right: expected value of optimal and target solution u

Number of random variables $= N = 1$

Number of Monte Carlo samples $= M = 100$



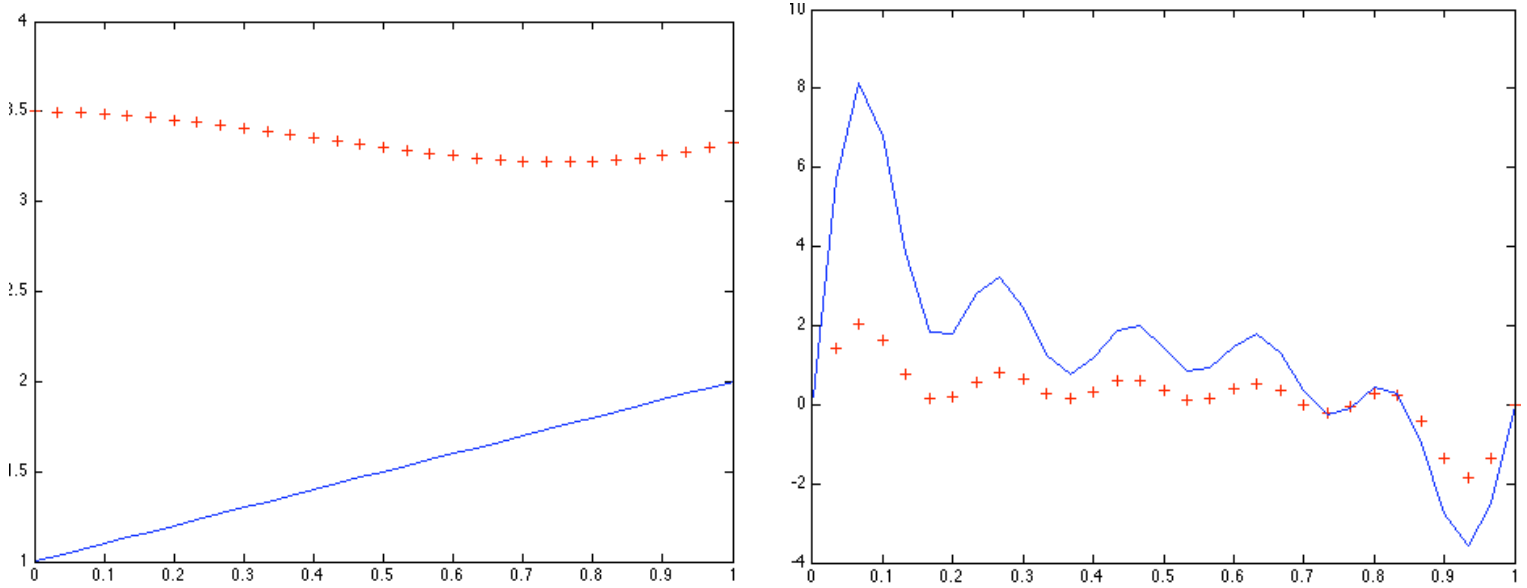


Left: expected value of optimal and target coefficient κ

Right: expected value of optimal and target solution u

Number of random variables $= N = 1$

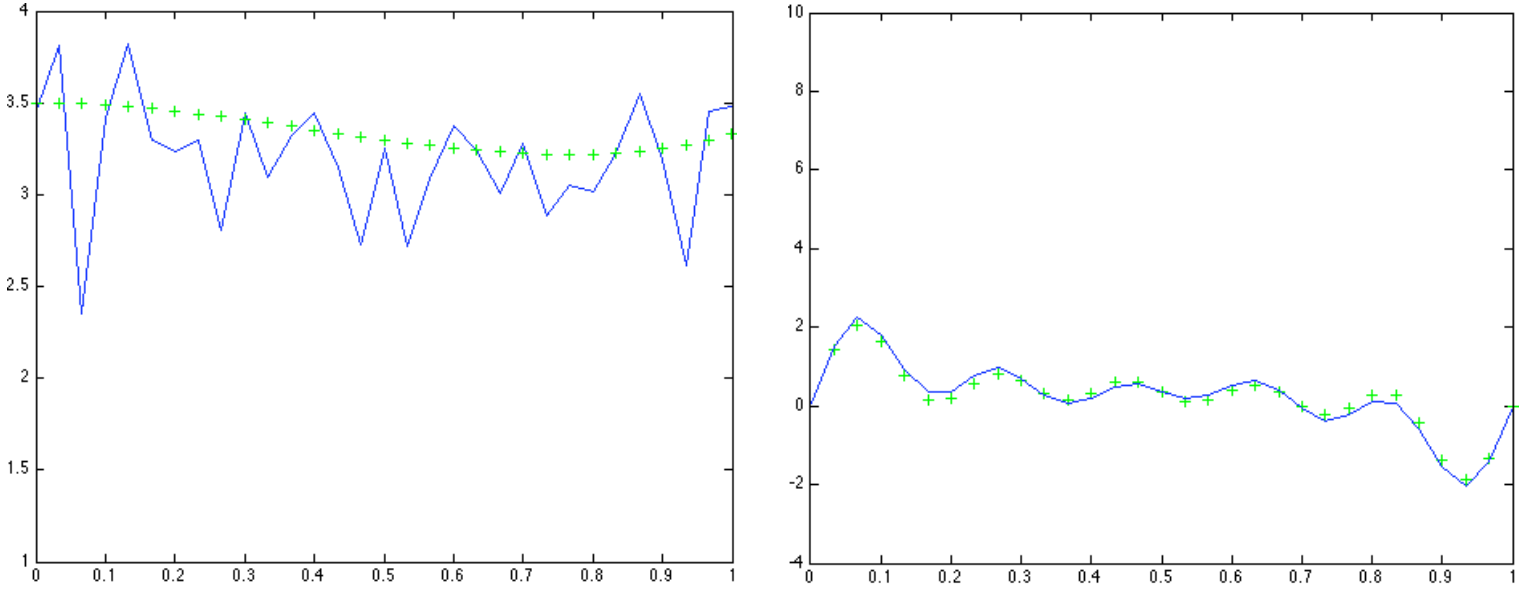
Number of anisotropic Smolyak collocation points $= M = 1$



Left: expected value of initial (blue) and target (red) coefficient κ

Right: expected value of initial and target solution u

Number of random variables $= N = 5$

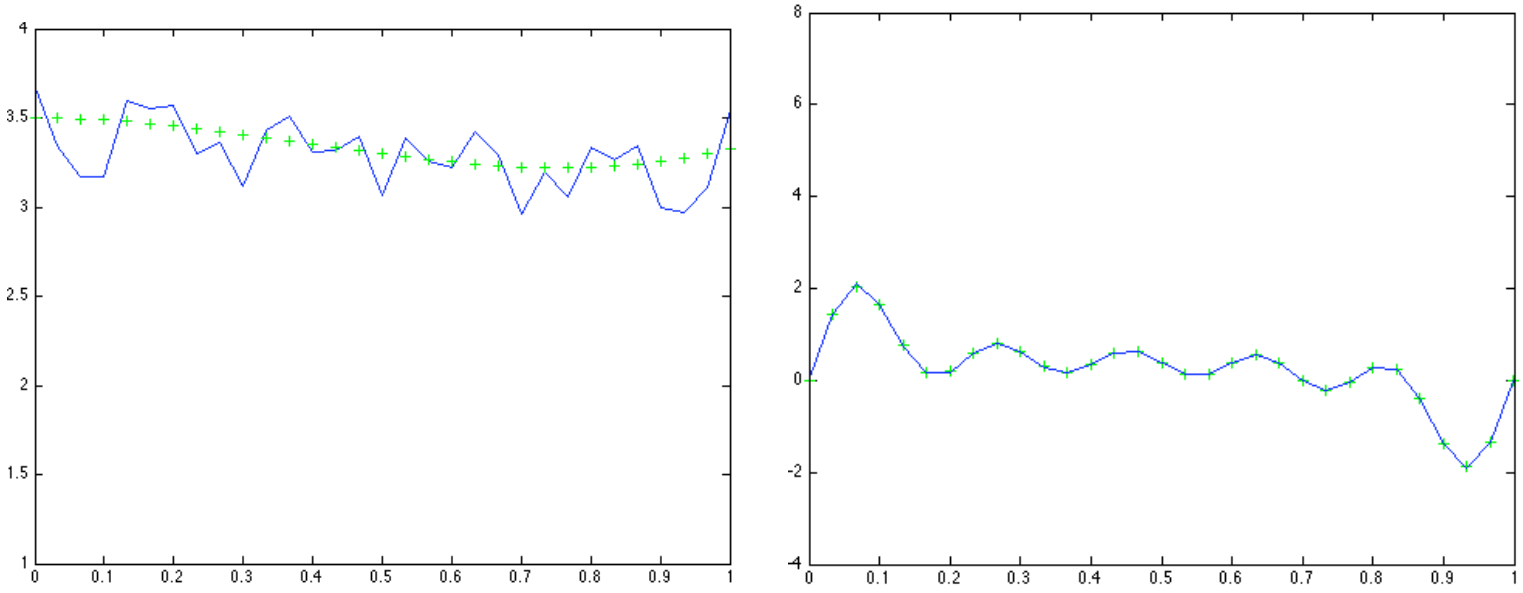


Left: expected value of optimal and target coefficient κ

Right: expected value of optimal and target solution u

Number of random variables $= N = 5$

Number of Monte Carlo samples $= M = 11$



Left: expected value of optimal and target coefficient κ

Right: expected value of optimal and target solution u

Number of random variables $= N = 5$

Number of anisotropic Smolyak collocation points $= M = 11$

N	MC	AS
5	7e+03	801
10	9e+06	1581
20	8e+09	11561

For N random parameters, the number of Monte Carlo samples and the number of anisotropic Smolyak collocation points required to reduce the original error in the expected values of both the solution u and coefficient κ by a factor of 10^6