Towards reduction of uncertainty in the operation of reservoir systems

Prof. Nathan L. Gibson
Department of Mathematics

Tseganeh Gichamo, Rachelle Valverde, Christopher H. Gifford-Miears and Arturo Leon
School of Civil and Construction Engineering

Oregon State University OSU

Reservoir System Modeling Technologies Conference
February 22, 2012

Outline

1. **Introduction**
   - Overview
   - Problem Setup

Outline

1. **Introduction**
   - Overview
   - Problem Setup
2. **Proposed Framework**
   - Preliminaries
   - Generalized Polynomial Chaos
   - Stochastic Collocation Method

Outline

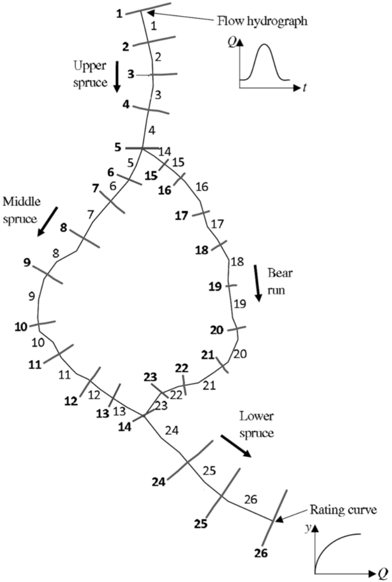
1. **Introduction**
   - Overview
   - Problem Setup
2. **Proposed Framework**
   - Preliminaries
   - Generalized Polynomial Chaos
   - Stochastic Collocation Method
3. **Numerical Results**
   - gPC Example

Outline

Outline

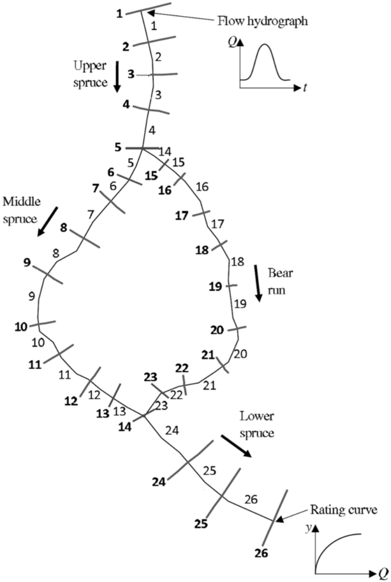
River Network System

- Input is flow hydrograph $Q(t)$
- River system flow dynamics determined by unsteady flow routing
- Nonlinear time-dependent system
- We use Performance Graphs approach to simulate (OSU Rivers)



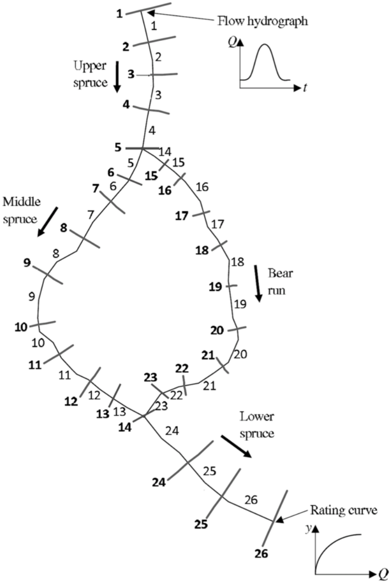
River Network System

- Input is flow hydrograph $Q(t)$
- River system flow dynamics determined by unsteady flow routing
- Nonlinear time-dependent system
- We use Performance Graphs approach to simulate (OSU Rivers)



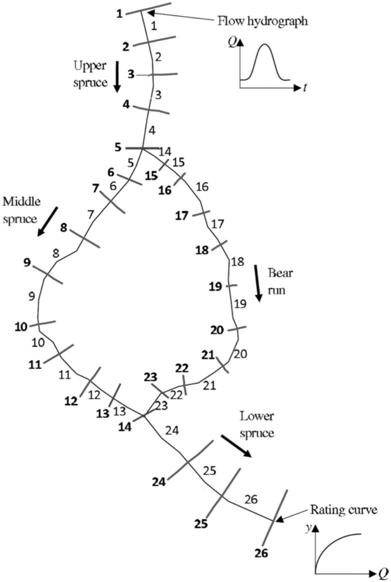
River Network System

- Input is flow hydrograph $Q(t)$
- River system flow dynamics determined by unsteady flow routing
- Nonlinear time-dependent system
- We use Performance Graphs approach to simulate (OSU Rivers)



River Network System

- Input is flow hydrograph $Q(t)$
- River system flow dynamics determined by unsteady flow routing
- Nonlinear time-dependent system
- We use Performance Graphs approach to simulate (OSU Rivers)



Incorporating Uncertainty

Incorporate uncertainty into inputs of the system

- We propose to add a stochastic component to input variables
- We use the system dynamics to propagate uncertainty
- This results in a stochastic representation of system variables
- Our approach allows for the explicit construction of the stochastic representation for select variables
  - other related quantities are implicitly stochastic

Incorporating Uncertainty

Incorporate uncertainty into inputs of the system

- We propose to add a stochastic component to input variables
- We use the system dynamics to propagate uncertainty
- This results in a stochastic representation of system variables
- Our approach allows for the explicit construction of the stochastic representation for select variables
  - other related quantities are implicitly stochastic

Incorporating Uncertainty

Incorporate uncertainty into inputs of the system

- We propose to add a stochastic component to input variables
- We use the system dynamics to propagate uncertainty
- This results in a stochastic representation of system variables
- Our approach allows for the explicit construction of the stochastic representation for select variables
    - other related quantities are implicitly stochastic

Incorporating Uncertainty

Incorporate uncertainty into inputs of the system

- We propose to add a stochastic component to input variables
- We use the system dynamics to propagate uncertainty
- This results in a stochastic representation of system variables
- Our approach allows for the explicit construction of the stochastic representation for select variables
  - other related quantities are implicitly stochastic

Incorporating Uncertainty

Incorporate uncertainty into inputs of the system

- We propose to add a stochastic component to input variables
- We use the system dynamics to propagate uncertainty
- This results in a stochastic representation of system variables
- Our approach allows for the explicit construction of the stochastic representation for select variables
  - other related quantities are implicitly stochastic

Advantages

Our proposed framework allows

- Simple extraction of statistics of solutions: mean, variance, etc.

Advantages

Our proposed framework allows

- Simple extraction of statistics of solutions: mean, variance, etc.
- Representation of stochastic outputs in the form of a polynomial in a random variable (easy to sample)

Advantages

Our proposed framework allows

- Simple extraction of statistics of solutions: mean, variance, etc.
- Representation of stochastic outputs in the form of a polynomial in a random variable (easy to sample)
    - to extract pdf

Advantages

Our proposed framework allows

- Simple extraction of statistics of solutions: mean, variance, etc.
- Representation of stochastic outputs in the form of a polynomial in a random variable (easy to sample)
    - to extract pdf
    - for Monte-Carlo methods

Advantages

Our proposed framework allows

- Simple extraction of statistics of solutions: mean, variance, etc.
- Representation of stochastic outputs in the form of a polynomial in a random variable (easy to sample)
  - to extract pdf
  - for Monte-Carlo methods
  - compute failure probabilities

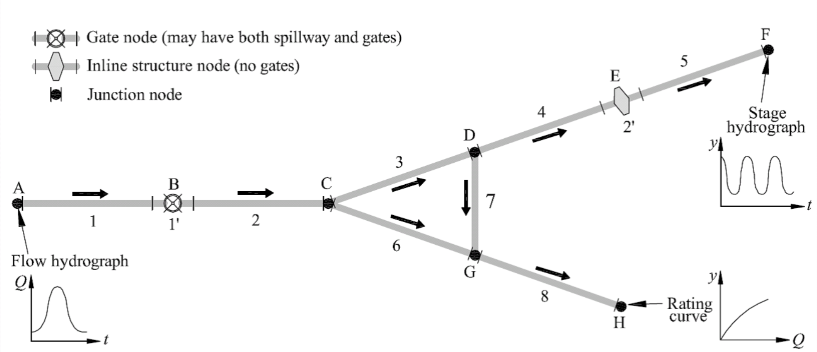
Advantages

Our proposed framework allows

- Simple extraction of statistics of solutions: mean, variance, etc.
- Representation of stochastic outputs in the form of a polynomial in a random variable (easy to sample)
  - to extract pdf
  - for Monte-Carlo methods
  - compute failure probabilities
- Polynomials computed using uncoupled, deterministic forward simulations with different parameters (embarrassingly parallel)

Simple Network System

Consider this simple network system



The (nonlinear) relationship between variables $\vec{X} = [y_{d_1}, \ldots, Q_{d_8}]$ is given by the following map

$$
\begin{bmatrix}
x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - \\
- & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - \\
- & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - \\
- & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - \\
- & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - \\
- & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - \\
- & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - \\
- & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x \\
- & - & - & - & - & - & - & - & - & x & - & - & - & - & - & - & x & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & x & - & - & x & - & - & - & x & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & x & - & - & x & - & - & - & x & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & x & - & - & - & - & - & - & x & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & - & - & - & x & - & - & - & - & - & x & x & - \\
- & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & x & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & x & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & x \\
- & x & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - \\
- & x & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - \\
- & - & x & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - \\
- & - & x & - & - & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x & - \\
- & - & - & - & - & x & - & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x \\
- & - & - & - & - & - & x & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & - & x \\
- & - & - & x & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & x & x & - & - & - \\
x & x & - & - & - & - & - & - & - & x & - & - & - & - & - & - & x & x & - & - & - & - & - & -
\end{bmatrix}
\begin{bmatrix}
y_{d_1} \\ y_{d_2} \\ y_{d_3} \\ y_{d_4} \\ y_{d_5} \\ y_{d_6} \\ y_{d_7} \\ y_{d_8} \\ Q_{u_1} \\ Q_{u_2} \\ Q_{u_3} \\ Q_{u_4} \\ Q_{u_5} \\ Q_{u_6} \\ Q_{u_7} \\ Q_{u_8} \\ Q_{d_1} \\ Q_{d_2} \\ Q_{d_3} \\ Q_{d_4} \\ Q_{d_5} \\ Q_{d_6} \\ Q_{d_7} \\ Q_{d_8}
\end{bmatrix}
=
\begin{bmatrix}
a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{15} \\ a_{16} \\ a_{17} \\ a_{18} \\ a_{19} \\ a_{20} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24}
\end{bmatrix}
$$

Problem Inputs

- Inputs: Q_{u_1}, y_5, Rating Curve, gate positions
- Assume uncertainty envelope around Q_{u_1} prediction (representing flow discharges upstream of reach 1)

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t)$$

- Additional input: pdf for $\tilde{Q}_{u_1}$

Problem Inputs

- Inputs: Q_{u_1}, y_5, Rating Curve, gate positions
- Assume uncertainty envelope around Q_{u_1} prediction (representing flow discharges upstream of reach 1)

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t)$$

- Additional input: pdf for $\tilde{Q}_{u_1}$

Problem Inputs

- Inputs: Q_{u_1}, y_5, Rating Curve, gate positions
- Assume uncertainty envelope around Q_{u_1} prediction (representing flow discharges upstream of reach 1)

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t)$$

- Additional input: pdf for $\tilde{Q}_{u_1}$

Questions one could ask

- What is the resulting pdf of Q_{d_8}?
  (propagation of uncertainty)
- What choice of gate positions cause Q_{d_8} to behave as desired?
  (on average)
- What choice of gate positions cause hydropower production $h(\vec{X})$ to behave as desired?
  (on average)
- What choice of gate positions minimize risk of flooding?
  (on average)
- Are there choices of gate positions that lead to robust predictions of Q_{d_8} or $h(\vec{X})$ or flood volumes?
  (minimize variance)

Questions one could ask

- What is the resulting pdf of Q_{d_8}?
  (propagation of uncertainty)
- What choice of gate positions cause Q_{d_8} to behave as desired?
  (on average)
- What choice of gate positions cause hydropower production $h(\vec{X})$ to behave as desired?
  (on average)
- What choice of gate positions minimize risk of flooding?
  (on average)
- Are there choices of gate positions that lead to robust predictions of Q_{d_8} or $h(\vec{X})$ or flood volumes?
  (minimize variance)

Questions one could ask

- What is the resulting pdf of Q_{d_8}?
  (propagation of uncertainty)
- What choice of gate positions cause Q_{d_8} to behave as desired?
  (on average)
- What choice of gate positions cause hydropower production $h(\vec{X})$ to behave as desired?
  (on average)
- What choice of gate positions minimize risk of flooding?
  (on average)
- Are there choices of gate positions that lead to robust predictions of Q_{d_8} or $h(\vec{X})$ or flood volumes?
  (minimize variance)

Questions one could ask

- What is the resulting pdf of Q_{d_8}?
  (propagation of uncertainty)
- What choice of gate positions cause Q_{d_8} to behave as desired?
  (on average)
- What choice of gate positions cause hydropower production $h(\vec{X})$ to behave as desired?
  (on average)
- What choice of gate positions minimize risk of flooding?
  (on average)
- Are there choices of gate positions that lead to robust predictions of Q_{d_8} or $h(\vec{X})$ or flood volumes?
  (minimize variance)

Questions one could ask

- What is the resulting pdf of Q_{d_8}?
  (propagation of uncertainty)
- What choice of gate positions cause Q_{d_8} to behave as desired?
  (on average)
- What choice of gate positions cause hydropower production $h(\vec{X})$ to behave as desired?
  (on average)
- What choice of gate positions minimize risk of flooding?
  (on average)
- Are there choices of gate positions that lead to robust predictions of Q_{d_8} or $h(\vec{X})$ or flood volumes?
  (minimize variance)

Outline

Mathematical Framework

Given

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t), \quad \tilde{Q}_{u_1} \sim F$$

Find gate positions $g(t)$ such that one or more of the following are minimized

Mathematical Framework

Given

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t), \quad \tilde{Q}_{u_1} \sim F$$

Find gate positions $g(t)$ such that one or more of the following are minimized

$$J_1 = \mathbb{E}\left[\|Q_{d_8} - d\|\right] = \int \left\| Q_{d_8}(\tilde{Q}_{u_1}) - d \right\| dF$$

Mathematical Framework

Given

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t), \quad \tilde{Q}_{u_1} \sim F$$

Find gate positions $g(t)$ such that one or more of the following are minimized

$$J_1 = \mathbb{E}\left[\| Q_{d_8} - d \|\right] = \int \left\| Q_{d_8}(\tilde{Q}_{u_1}) - d \right\| dF$$

not the same as $\|\mathbb{E}[Q_{d_8}] - d\|$

Mathematical Framework

Given

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t), \quad \tilde{Q}_{u_1} \sim F$$

Find gate positions $g(t)$ such that one or more of the following are minimized

$$J_1 = \mathbb{E}\left[\| Q_{d_8} - d \|\right] = \int \left\| Q_{d_8}(\tilde{Q}_{u_1}) - d \right\| dF$$

not the same as $\|\mathbb{E}[Q_{d_8}] - d\|$

$$J_2 = Var(Q_{d_8})$$

Mathematical Framework

Given

$$Q_{u_1}(t) = \overline{Q}_{u_1}(t) + \tilde{Q}_{u_1}(t), \quad \tilde{Q}_{u_1} \sim F$$

Find gate positions $g(t)$ such that one or more of the following are minimized

$$J_1 = \mathbb{E}\left[\|Q_{d_8} - d\|\right] = \int \left\| Q_{d_8}(\tilde{Q}_{u_1}) - d \right\| dF$$

not the same as $\|\mathbb{E}[Q_{d_8}] - d\|$

$$J_2 = Var(Q_{d_8})$$

$$J_3 = Prob[Q_{d_8} < \text{tolerance}]$$

or similar for $\vec{X}_i$ or $h(\vec{X})$ or FV.

Requirements

- Need a framework that allows fast computation of statistics and/or failure probabilities of select variables in the solution of the system. (or nonlinear/non-smooth functions of select variables)
- To take advantage of an already developed flow dynamics model based on the HPG/VPG approach, we will incorporate the proposed uncertainty framework non-intrusively into OSU Rivers.

Requirements

- Need a framework that allows fast computation of statistics and/or failure probabilities of select variables in the solution of the system. (or nonlinear/non-smooth functions of select variables)

- To take advantage of an already developed flow dynamics model based on the HPG/VPG approach, we will incorporate the proposed uncertainty framework non-intrusively into OSU Rivers.

Requirements

- Need a framework that allows fast computation of statistics and/or failure probabilities of select variables in the solution of the system. (or nonlinear/non-smooth functions of select variables)
- To take advantage of an already developed flow dynamics model based on the HPG/VPG approach, we will incorporate the proposed uncertainty framework non-intrusively into OSU Rivers.

Generalized Polynomial Chaos

- Our approach is to explicitly model the random space (via random variables and processes) and perform a generalized Polynomial Chaos (gPC) representation
- This method uses an orthogonal polynomial expansion in random space to represent the stochastic input quantities as well as the solutions to the system.
- Convergence of polynomial chaos methods can be shown to be exponential in the number of basis functions.

Generalized Polynomial Chaos

- Our approach is to explicitly model the random space (via random variables and processes) and perform a generalized Polynomial Chaos (gPC) representation
- This method uses an orthogonal polynomial expansion in random space to represent the stochastic input quantities as well as the solutions to the system.
- Convergence of polynomial chaos methods can be shown to be exponential in the number of basis functions.

Generalized Polynomial Chaos

- Our approach is to explicitly model the random space (via random variables and processes) and perform a generalized Polynomial Chaos (gPC) representation
- This method uses an orthogonal polynomial expansion in random space to represent the stochastic input quantities as well as the solutions to the system.
- Convergence of polynomial chaos methods can be shown to be exponential in the number of basis functions.

Example

To illustrate the idea of our approach, we present the following example. Consider the quantity Q_{u_1} above.

- We may assume that the uncertainty around the prediction $\overline{Q}_{u_1}(t)$ is relative with magnitude 10%.
- We introduce the standard random variable (hereafter referred to as a *germ*) $\xi \sim Beta(\alpha, \beta)$ with support $[-1, 1]$ then

$$Q_{u_1} = \overline{Q}_{u_1} + 0.1\xi\overline{Q}_{u_1} \tag{1}$$

- Equation (1) represents a polynomial chaos expansion of the random input.
- If $\alpha = \beta$, F is a symmetric beta distribution centered around mean 0; for the special case $\alpha = \beta = 0$ this is simply a uniform distribution.

Example

To illustrate the idea of our approach, we present the following example. Consider the quantity Q_{u_1} above.

- We may assume that the uncertainty around the prediction $\overline{Q}_{u_1}(t)$ is relative with magnitude 10%.
- We introduce the standard random variable (hereafter referred to as a *germ*) $\xi \sim Beta(\alpha, \beta)$ with support $[-1, 1]$ then

$$Q_{u_1} = \overline{Q}_{u_1} + 0.1\xi\overline{Q}_{u_1} \quad (1)$$

- Equation (1) represents a polynomial chaos expansion of the random input.
- If $\alpha = \beta$, F is a symmetric beta distribution centered around mean 0; for the special case $\alpha = \beta = 0$ this is simply a uniform distribution.

Example

To illustrate the idea of our approach, we present the following example. Consider the quantity Q_{u_1} above.

- We may assume that the uncertainty around the prediction $\overline{Q}_{u_1}(t)$ is relative with magnitude 10%.
- We introduce the standard random variable (hereafter referred to as a *germ*) $\xi \sim Beta(\alpha, \beta)$ with support $[-1, 1]$ then

$$Q_{u_1} = \overline{Q}_{u_1} + 0.1\xi\overline{Q}_{u_1} \tag{1}$$

- Equation (1) represents a polynomial chaos expansion of the random input.
- If $\alpha = \beta$, F is a symmetric beta distribution centered around mean 0; for the special case $\alpha = \beta = 0$ this is simply a uniform distribution.

Example

To illustrate the idea of our approach, we present the following example. Consider the quantity Q_{u_1} above.

- We may assume that the uncertainty around the prediction $\overline{Q}_{u_1}(t)$ is relative with magnitude 10%.
- We introduce the standard random variable (hereafter referred to as a *germ*) $\xi \sim Beta(\alpha, \beta)$ with support $[-1, 1]$ then

$$Q_{u_1} = \overline{Q}_{u_1} + 0.1\xi\overline{Q}_{u_1} \tag{1}$$

- Equation (1) represents a polynomial chaos expansion of the random input.
- If $\alpha = \beta$, F is a symmetric beta distribution centered around mean 0; for the special case $\alpha = \beta = 0$ this is simply a uniform distribution.

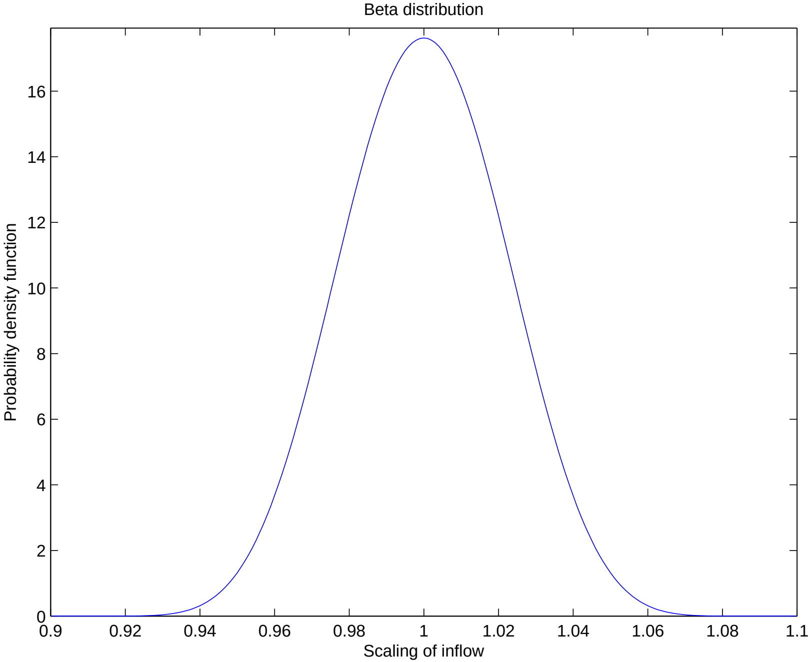
Example

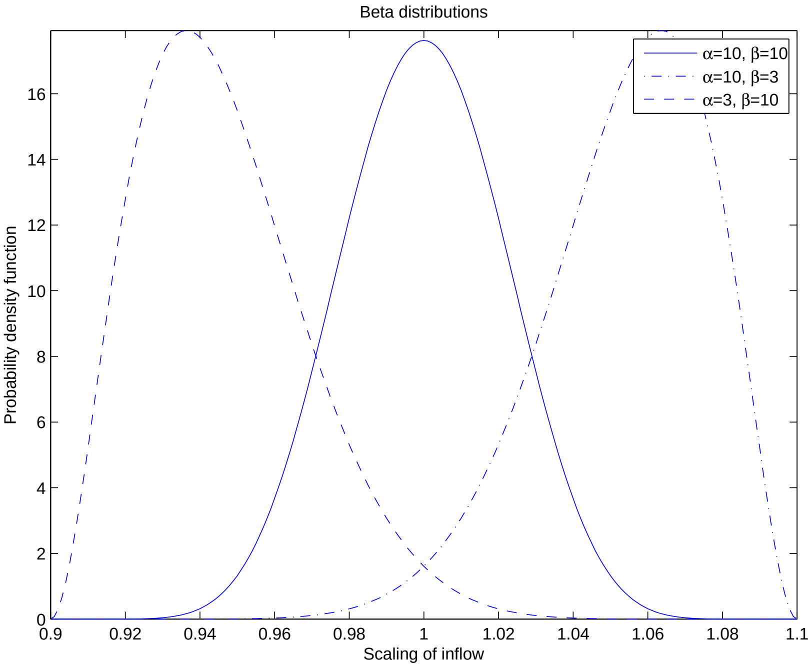
To illustrate the idea of our approach, we present the following example. Consider the quantity Q_{u_1} above.

- We may assume that the uncertainty around the prediction $\overline{Q}_{u_1}(t)$ is relative with magnitude 10%.
- We introduce the standard random variable (hereafter referred to as a *germ*) $\xi \sim Beta(\alpha, \beta)$ with support $[-1, 1]$ then

$$Q_{u_1} = \overline{Q}_{u_1} + 0.1\xi\overline{Q}_{u_1} \tag{1}$$

- Equation (1) represents a polynomial chaos expansion of the random input.
- If $\alpha = \beta$, F is a symmetric beta distribution centered around mean 0; for the special case $\alpha = \beta = 0$ this is simply a uniform distribution.


Beta distribution
Probability density function
0
2
4
6
8
10
12
14
16
0.9
0.92
0.94
0.96
0.98
1
1.02
1.04
1.06
1.08
1.1
Scaling of inflow



Beta distributions
α=10, β=10
α=10, β=3
α=3, β=10
Probability density function
Scaling of inflow
0
2
4
6
8
10
12
14
16
0.9
0.92
0.94
0.96
0.98
1
1.02
1.04
1.06
1.08
1.1

Generalizations

- Continuous distributions that can be handled directly: Uniform, Beta, Gaussian, Gamma
- Discrete distributions: Poisson, Binomial, Hypergeometric
- Choice of distribution determines corresponding orthogonal polynomials (e.g., Gaussian pdf defines Hermite orthogonality)
- Other distributions require non-linear transformations of the above, or manual construction of orthogonal polynomials
- Random inputs can be random variables or random processes (time-dependent), e.g., represented by a Karhunen-Loeve (KL) expansion
- Any number of independent random inputs may be used, each with their own distribution

Generalizations

- Continuous distributions that can be handled directly: Uniform, Beta, Gaussian, Gamma
- Discrete distributions: Poisson, Binomial, Hypergeometric
- Choice of distribution determines corresponding orthogonal polynomials (e.g., Gaussian pdf defines Hermite orthogonality)
- Other distributions require non-linear transformations of the above, or manual construction of orthogonal polynomials
- Random inputs can be random variables or random processes (time-dependent), e.g., represented by a Karhunen-Loeve (KL) expansion
- Any number of independent random inputs may be used, each with their own distribution

Generalizations

- Continuous distributions that can be handled directly: Uniform, Beta, Gaussian, Gamma
- Discrete distributions: Poisson, Binomial, Hypergeometric
- Choice of distribution determines corresponding orthogonal polynomials (e.g., Gaussian pdf defines Hermite orthogonality)
- Other distributions require non-linear transformations of the above, or manual construction of orthogonal polynomials
- Random inputs can be random variables or random processes (time-dependent), e.g., represented by a Karhunen-Loeve (KL) expansion
- Any number of independent random inputs may be used, each with their own distribution

Generalizations

- Continuous distributions that can be handled directly: Uniform, Beta, Gaussian, Gamma
- Discrete distributions: Poisson, Binomial, Hypergeometric
- Choice of distribution determines corresponding orthogonal polynomials (e.g., Gaussian pdf defines Hermite orthogonality)
- Other distributions require non-linear transformations of the above, or manual construction of orthogonal polynomials
- Random inputs can be random variables or random processes (time-dependent), e.g., represented by a Karhunen-Loeve (KL) expansion
- Any number of independent random inputs may be used, each with their own distribution

Generalizations

- Continuous distributions that can be handled directly: Uniform, Beta, Gaussian, Gamma
- Discrete distributions: Poisson, Binomial, Hypergeometric
- Choice of distribution determines corresponding orthogonal polynomials (e.g., Gaussian pdf defines Hermite orthogonality)
- Other distributions require non-linear transformations of the above, or manual construction of orthogonal polynomials
- Random inputs can be random variables or random processes (time-dependent), e.g., represented by a Karhunen-Loeve (KL) expansion
- Any number of independent random inputs may be used, each with their own distribution

Generalizations

- Continuous distributions that can be handled directly: Uniform, Beta, Gaussian, Gamma
- Discrete distributions: Poisson, Binomial, Hypergeometric
- Choice of distribution determines corresponding orthogonal polynomials (e.g., Gaussian pdf defines Hermite orthogonality)
- Other distributions require non-linear transformations of the above, or manual construction of orthogonal polynomials
- Random inputs can be random variables or random processes (time-dependent), e.g., represented by a Karhunen-Loeve (KL) expansion
- Any number of independent random inputs may be used, each with their own distribution

Stochastic Galerkin Method

- In a Galerkin method, we seek to determine the coefficients of a gPC expansion of each component of the solution vector

$$\vec{X} = [y_{d_1}, \ldots, Q_{d_8}]$$

- To do this, one may take a Galerkin projection of the original system, but with these expansions substituted in for the solution quantities
- The resulting integrals can be computed analytically due to the polynomial basis representation
- This approach in general leads to a large coupled system of equations for the gPC coefficients
  (e.g., intrusive method: changes the system to be solved)
- This new system must be discretized in space and time

Stochastic Galerkin Method

- In a Galerkin method, we seek to determine the coefficients of a gPC expansion of each component of the solution vector

$$\vec{X} = [y_{d_1}, \ldots, Q_{d_8}]$$

- To do this, one may take a Galerkin projection of the original system, but with these expansions substituted in for the solution quantities
- The resulting integrals can be computed analytically due to the polynomial basis representation
- This approach in general leads to a large coupled system of equations for the gPC coefficients
  (e.g., intrusive method: changes the system to be solved)
- This new system must be discretized in space and time

Stochastic Galerkin Method

- In a Galerkin method, we seek to determine the coefficients of a gPC expansion of each component of the solution vector

$$\vec{X} = [y_{d_1}, \ldots, Q_{d_8}]$$

- To do this, one may take a Galerkin projection of the original system, but with these expansions substituted in for the solution quantities
- The resulting integrals can be computed analytically due to the polynomial basis representation
- This approach in general leads to a large coupled system of equations for the gPC coefficients
  (e.g., intrusive method: changes the system to be solved)
- This new system must be discretized in space and time

Stochastic Galerkin Method

- In a Galerkin method, we seek to determine the coefficients of a gPC expansion of each component of the solution vector

$$\vec{X} = [y_{d_1}, \ldots, Q_{d_8}]$$

- To do this, one may take a Galerkin projection of the original system, but with these expansions substituted in for the solution quantities
- The resulting integrals can be computed analytically due to the polynomial basis representation
- This approach in general leads to a large coupled system of equations for the gPC coefficients
  (e.g., intrusive method: changes the system to be solved)
- This new system must be discretized in space and time

Stochastic Galerkin Method

- In a Galerkin method, we seek to determine the coefficients of a gPC expansion of each component of the solution vector

$$\vec{X} = [y_{d_1}, \ldots, Q_{d_8}]$$

- To do this, one may take a Galerkin projection of the original system, but with these expansions substituted in for the solution quantities
- The resulting integrals can be computed analytically due to the polynomial basis representation
- This approach in general leads to a large coupled system of equations for the gPC coefficients
  (e.g., intrusive method: changes the system to be solved)
- This new system must be discretized in space and time

Stochastic Collocation Method

- An alternative is to numerically approximate the integrals
- Consider Q_{d_8}: its representation in terms of a degree P expansion

$$Q_{d_8}^P(t,\xi) = \sum_{i=0}^{P} v_i(t)\phi_i(\xi)$$

  where $\phi_i(\xi)$ are the basis functions
  (Jacobi polynomials in the case of a Beta distribution of inputs).
- Each gPC expansion coefficient is given by

$$v_i(t) = \mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)]$$

  i.e., the expected value with respect to F of the true solution times the (normalized) basis function

Stochastic Collocation Method

- An alternative is to numerically approximate the integrals
- Consider Q_{d_8}: its representation in terms of a degree P expansion

$$Q_{d_8}^P(t,\xi) = \sum_{i=0}^{P} v_i(t)\phi_i(\xi)$$

where $\phi_i(\xi)$ are the basis functions
(Jacobi polynomials in the case of a Beta distribution of inputs).

- Each gPC expansion coefficient is given by

$$v_i(t) = \mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)]$$

i.e., the expected value with respect to F of the true solution times the (normalized) basis function

Stochastic Collocation Method

- An alternative is to numerically approximate the integrals
- Consider Q_{d_8}: its representation in terms of a degree P expansion

$$Q_{d_8}^P(t,\xi) = \sum_{i=0}^{P} v_i(t)\phi_i(\xi)$$

where $\phi_i(\xi)$ are the basis functions
(Jacobi polynomials in the case of a Beta distribution of inputs).

- Each gPC expansion coefficient is given by

$$v_i(t) = \mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)]$$

i.e., the expected value with respect to F of the true solution times the (normalized) basis function

Stochastic Collocation Method with Gaussian Quadrature

- The computation of the integral in

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] = \int Q_{d_8}(t,\xi)\phi_i(\xi)dF(\xi)$$

can be performed efficiently via Gaussian quadrature.

- Gaussian quadrature applies to functions which can be represented as $g(\xi)W(\xi)$ where $g(\xi)$ is well-approximated by a polynomial.
- Then the nodes ξ_j of the quadrature rule are the roots of an orthogonal polynomial in the support of F

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j Q_{d_8}(t,\xi_j)\phi_i(\xi_j)$$

- Stochastic Collocation thus requires only deterministic system solutions for the fixed values $\{\xi_j\}_{j=1}^{N}$ of the random variable ξ

Stochastic Collocation Method with Gaussian Quadrature

- The computation of the integral in

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] = \int Q_{d_8}(t,\xi)\phi_i(\xi)dF(\xi)$$

  can be performed efficiently via Gaussian quadrature.
- Gaussian quadrature applies to functions which can be represented as $g(\xi)W(\xi)$ where $g(\xi)$ is well-approximated by a polynomial.
- Then the nodes ξ_j of the quadrature rule are the roots of an orthogonal polynomial in the support of F

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j Q_{d_8}(t,\xi_j)\phi_i(\xi_j)$$

- Stochastic Collocation thus requires only deterministic system solutions for the fixed values $\{\xi_j\}_{j=1}^{N}$ of the random variable ξ

Stochastic Collocation Method with Gaussian Quadrature

- The computation of the integral in

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] = \int Q_{d_8}(t,\xi)\phi_i(\xi)dF(\xi)$$

can be performed efficiently via Gaussian quadrature.

- Gaussian quadrature applies to functions which can be represented as $g(\xi)W(\xi)$ where $g(\xi)$ is well-approximated by a polynomial.
- Then the nodes ξ_j of the quadrature rule are the roots of an orthogonal polynomial in the support of F

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j Q_{d_8}(t,\xi_j)\phi_i(\xi_j)$$

- Stochastic Collocation thus requires only deterministic system solutions for the fixed values $\{\xi_j\}_{j=1}^N$ of the random variable ξ

Stochastic Collocation Method with Gaussian Quadrature

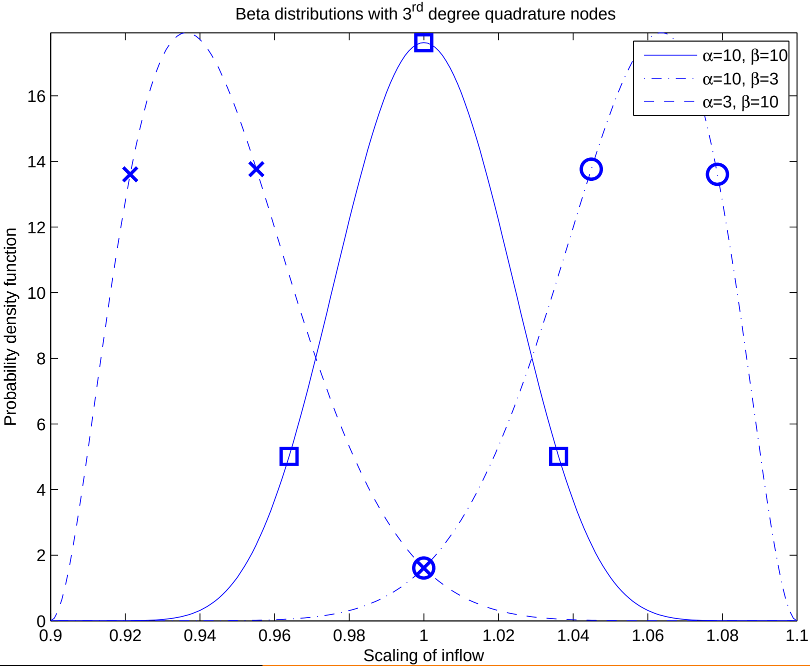
- The computation of the integral in

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] = \int Q_{d_8}(t,\xi)\phi_i(\xi)dF(\xi)$$

  can be performed efficiently via Gaussian quadrature.
- Gaussian quadrature applies to functions which can be represented as $g(\xi)W(\xi)$ where $g(\xi)$ is well-approximated by a polynomial.
- Then the nodes ξ_j of the quadrature rule are the roots of an orthogonal polynomial in the support of F

$$\mathbb{E}[Q_{d_8}(t,\xi)\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j Q_{d_8}(t,\xi_j)\phi_i(\xi_j)$$

- Stochastic Collocation thus requires only deterministic system solutions for the fixed values $\{\xi_j\}_{j=1}^{N}$ of the random variable ξ


Beta distributions with 3rd degree quadrature nodes
α=10, β=10
α=10, β=3
α=3, β=10
Probability density function
Scaling of inflow
0
2
4
6
8
10
12
14
16
0.9
0.92
0.94
0.96
0.98
1
1.02
1.04
1.06
1.08
1.1

Comments on Stochastic Collocation

- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Comments on Stochastic Collocation

- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Comments on Stochastic Collocation

- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Comments on Stochastic Collocation

- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Comments on Stochastic Collocation

- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Comments on Stochastic Collocation

- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Comments on Stochastic Collocation

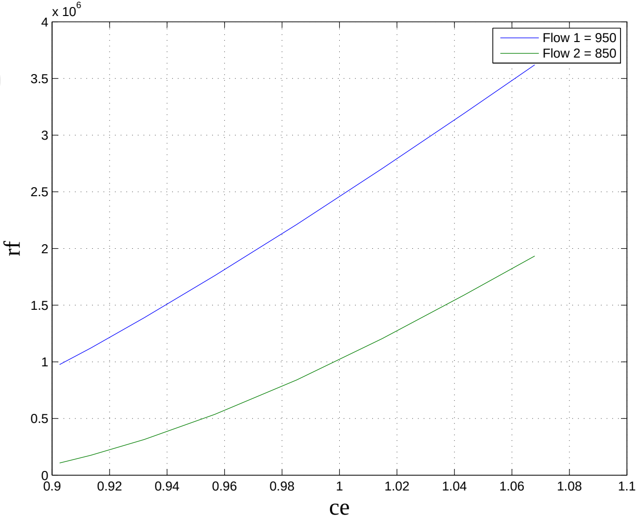
- System solutions at ξ_j can be recycled even if the input pdf changes
- Gaussian nodes are pre-determined (by choice of distribution) and thus simulations are independent and parallelizable!
- For large random dimension, sparse grids (Smolyak) are used to mitigate curse of dimensionality in quadrature
- Implemented in DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit by Sandia National Laboratories
  - Open source, C++ software
  - Extensible interface between simulation codes and various uncertainty quantification methods
- After simulations are performed, gPC expansion for any function of output may be easily constructed

$$\mathbb{E}[f(\vec{X}(t,\xi))\phi_i(\xi)] \approx \sum_{j=1}^{N} w_j f(\vec{X}(t,\xi_j))\phi_i(\xi_j)$$

Outline

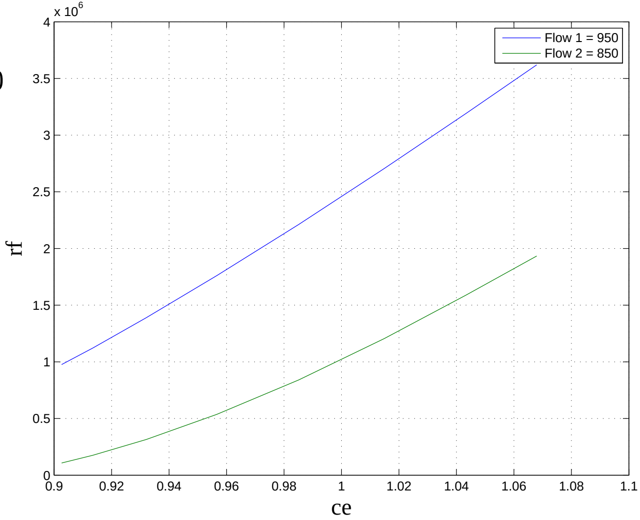
We build the gPC expansion (response function) for the flood volume in a single reach simulation using two different predictions of Q_{u_1}:

- Peak flows of 850 and 950
- 10% relative Beta uncertainty envelope
- FV given by FPGs
- Expansion is non-linear (yet low degree)



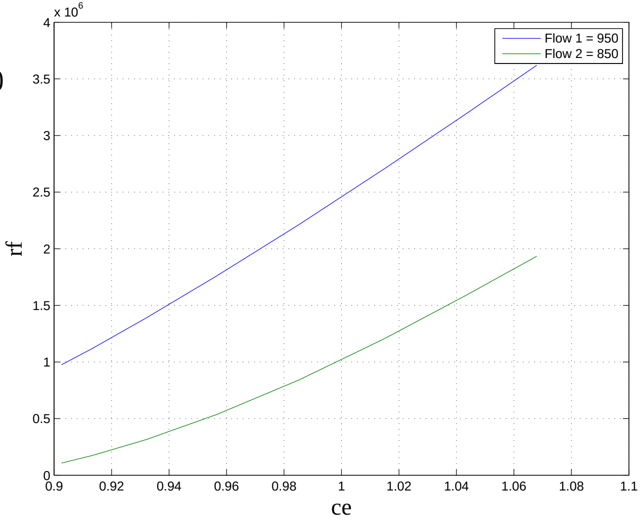
We build the gPC expansion (response function) for the flood volume in a single reach simulation using two different predictions of Q_{u_1}:

- Peak flows of 850 and 950
- 10% relative Beta uncertainty envelope
- FV given by FPGs
- Expansion is non-linear (yet low degree)



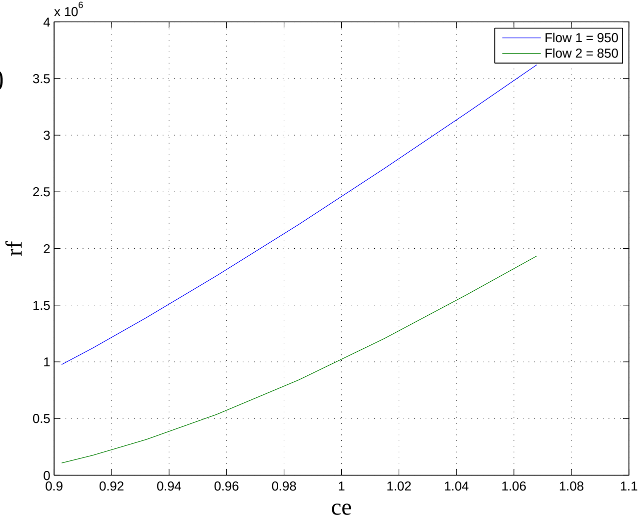
We build the gPC expansion (response function) for the flood volume in a single reach simulation using two different predictions of Q_{u_1}:

- Peak flows of 850 and 950
- 10% relative Beta uncertainty envelope
- FV given by FPGs
- Expansion is non-linear (yet low degree)



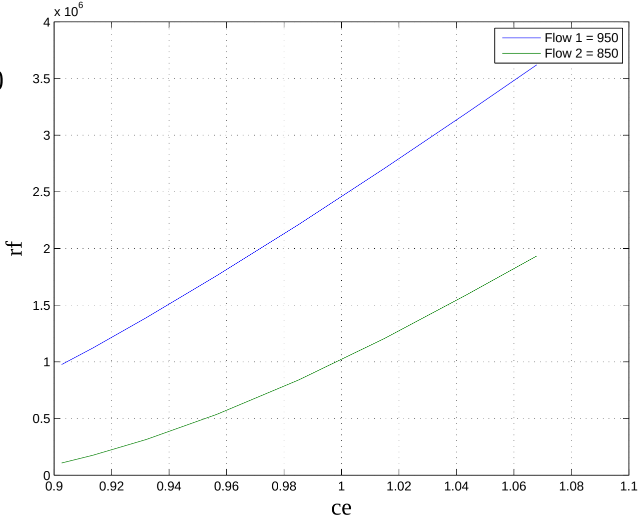
We build the gPC expansion (response function) for the flood volume in a single reach simulation using two different predictions of Q_{u_1}:

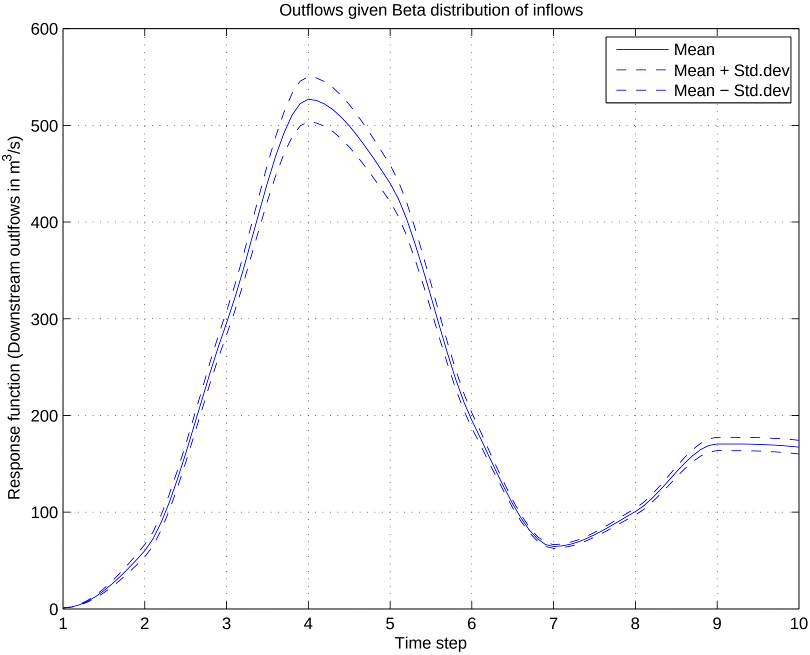
- Peak flows of 850 and 950
- 10% relative Beta uncertainty envelope
- FV given by FPGs
- Expansion is non-linear (yet low degree)

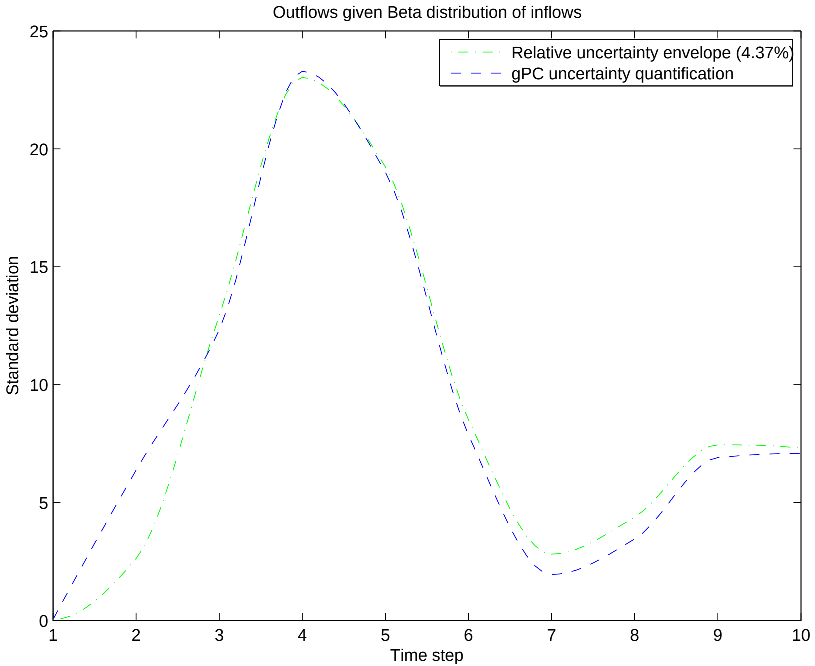


We build the gPC expansion (response function) for the flood volume in a single reach simulation using two different predictions of Q_{u_1}:

- Peak flows of 850 and 950
- 10% relative Beta uncertainty envelope
- FV given by FPGs
- Expansion is non-linear (yet low degree)




Outflows given Beta distribution of inflows
Mean
Mean + Std.dev
Mean − Std.dev
Response function (Downstream outflows in m3/s)
Time step
600
500
400
300
200
100
0
1
2
3
4
5
6
7
8
9
10


Outflows given Beta distribution of inflows
Relative uncertainty envelope (4.37%)
gPC uncertainty quantification
Standard deviation
Time step
0
5
10
15
20
25
1
2
3
4
5
6
7
8
9
10

Outline

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)

Future Work

- Demonstrate on large-scale example
- Incorporate into Optimal Control Framework
  - OSU Rivers: NSGA-II Genetic Algorithm
  - Semi-smooth Newton framework (Primal-Dual Active Set Strategy)
  - Hybrid approach (local vs global)
- Random Optimal Control
  - Use Bayesian Inference framework to determine gPC for optimal control
  - Polynomial representation allows pdf of control to be determined
  - Allows for quantification of robustness of control relative to uncertainty (distributional sensitivities)
  - Likelihood of control meeting objectives (conditional probabilities)