A simple implementation of polynomial chaos expansions

Jonathan Feinberg

Simula Research Laboratory AS

14th May 2014

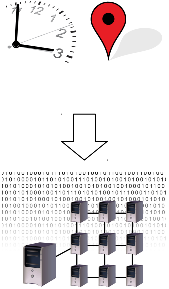
Overview of the three parts of the talk.

1. Problem Description
2. Theory
3. Implementation

Description of a general model with uncertain components.

Space x, time t and unknown model parameters q with density p_q

Description of a general model with uncertain components.



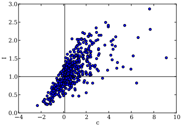
Space x, time t and unknown model parameters q with density p_q

(Costly) numerical solver $U = u(x, t; q)$

Description of a general model with uncertain components.

Space x, time t and unknown model parameters q with density p_q

(Costly) numerical solver $U = u(x, t; q)$



Model analysis $V(U)$

Model approximation using polynomial chaos expansion.

Assumption:

$u(x, t; q)$ is smooth as a function of q

Model approximation using polynomial chaos expansion.

Assumption:

$u(x, t; q)$ is smooth as a function of q

Proposal:

Approximate u with a polynomial:

$$u(x,t;q) \quad \approx \quad \hat{u}(x,t;q) = \sum_{i=0}^{N} \underbrace{c_i(x,t)}_{\text{coefficient}} \quad \underbrace{\Phi_i(q)}_{\text{polynomial}}$$

Model approximation using polynomial chaos expansion.

Assumption:

$u(x, t; q)$ is smooth as a function of q

Proposal:

Approximate u with a polynomial:

$$u(x,t;q) \quad \approx \quad \hat{u}(x,t;q) = \sum_{i=0}^{N} \underbrace{c_i(x,t)}_{\text{coefficient}} \quad \underbrace{\Phi_i(q)}_{\text{polynomial}}$$

Choose polynomials to be orthogonal in custom inner product

$$\langle \Phi_i, \Phi_j \rangle = \mathrm{E}[\Phi_i \Phi_j] = 0 \qquad\qquad i \neq j$$

Advantageous of using a polynomial chaos expansions.

- Scalable accuracy

Advantageous of using a polynomial chaos expansions.

- Scalable accuracy
- Cheap model analysis

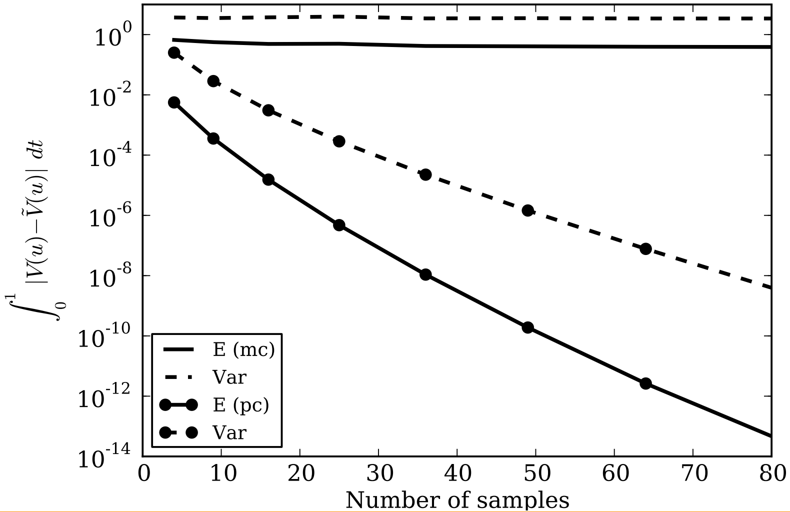
Advantageous of using a polynomial chaos expansions.

- Scalable accuracy
- Cheap model analysis
- Optimality linked to statistical property

Advantageous of using a polynomial chaos expansions.

- Scalable accuracy
- Cheap model analysis
- Optimality linked to statistical property
- Coefficients c_i are Fourier

Polynomial chaos often outperforms other methods with several orders of magnitude.



Example

```
import chaospy as cp

# Uncertain model parameters
dist_c = cp.Uniform(2, 10)
dist_I = cp.Normal(2, 2)
dist = cp.J(c,I)
```

Example

```
import chaospy as cp

# Uncertain model parameters
dist_c = cp.Uniform(2, 10)
dist_I = cp.Normal(2, 2)
dist = cp.J(c,I)

def model_solver(z):
    c,I = z
    t = linspace(0, 1, 100)
    return I*e**(-c*t)
```

Example

```
import chaospy as cp

# Uncertain model parameters
dist_c = cp.Uniform(2, 10)
dist_I = cp.Normal(2, 2)
dist = cp.J(c,I)

def model_solver(z):
    c,I = z
    t = linspace(0, 1, 100)
    return I*e**(-c*t)

# Implementaiton of probabilistic collocation method
Q = cp.pcm(func, 2, dist)
# Number of evaluations: 9
```

Example

```
import chaospy as cp

# Uncertain model parameters
dist_c = cp.Uniform(2, 10)
dist_I = cp.Normal(2, 2)
dist = cp.J(c,I)

def model_solver(z):
    c,I = z
    t = linspace(0, 1, 100)
    return I*e**(-c*t)

# Implementaiton of probabilistic collocation method
Q = cp.pcm(func, 2, dist)
# Number of evaluations: 9

# Analysis
print cp.E(Q, Z)
# [ 2.          1.88290007  1.77362073   ...
print cp.Var(Q, Z)
# [ 4.          3.54780499  3.15456747   ...
```

Conclusion

- Easy setup
- Few assumptions
- Fast convergence
- Functional representation

Conclusion

- Easy setup
- Few assumptions
- Fast convergence
- Functional representation

Thank you!

Calculating Fourier coefficients c_i using numerical integration.

$$
\begin{aligned}
c_i = \tfrac{\langle u, \Phi_i \rangle}{\|\Phi_i\|^2} &= \frac{\mathrm{E}[u\Phi_i]}{\mathrm{E}\left[\Phi_i^2\right]} \\
&= \frac{1}{\mathrm{E}\left[\Phi_i^2\right]} \int u(x,t;q)\Phi_i(q)p_q(q)\,\mathrm{d}q \\
&\approx \frac{1}{\mathrm{E}\left[\Phi_i^2\right]} \sum_{k=0}^{K} u(x,t;q_k)\Phi_i(q_k)
\end{aligned}
$$

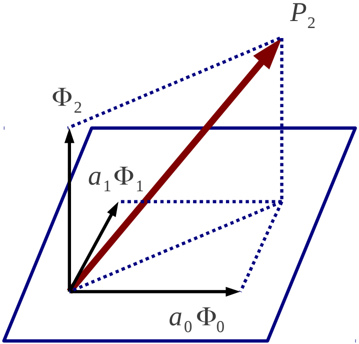
Constructing Orthogonal Polynomials Φ_i using orthogonalization methods.

Gram-Schmidt orthogonalization:

$$\Phi_0 = P_0$$

$$\Phi_n = P_n - \sum_{i=0}^{n-1} \frac{\langle P_n, \Phi_i \rangle}{\|\Phi_i\|^2} \Phi_i$$

$$= P_n - \sum_{i=0}^{n-1} \frac{\mathrm{E}[P_n \Phi_i]}{\mathrm{E}[P_n^2]} \Phi_i$$



P_i any linear independent basis.