Orthogonal Polynomials

Uncertainty Quantification Using Polynomial Chaos

Grey Yuan

Supervised by:
Prof. Roni Khazaka

McGill

SURE Project

23 July 2021

Introduction

- Growth of very-large scale integration(VLSI)
- Systems need to be compact while maintaining their precision
- Unpredictability leads to numerical uncertainty
- Uncertainty quantification to predict statistical distribution
  - Traditional Monte Carlo (MC) Method
  - Polynomial Chaos (PC) Method

McGill

Table of Content

McGill

1-dimensional collocation matrix

- Hermite(probalists) recurrence relationship:

$$H_0(\xi) = 1 \quad H_1(\xi) = \xi \quad H_{n+1}(\xi) = \xi H_n(\xi) - nH_{n-1}(\xi)$$

- Example terms:

$$\begin{aligned} He_0(x) &= 1, \\ He_1(x) &= x, \\ He_2(x) &= x^2 - 1, \\ He_3(x) &= x^3 - 3x, \end{aligned}$$

- API: object.HermiteCollocationMatrix(x)
  - object: an orthogonal polynomial object with order as a property
  - x corresponds to the sample point(s)

McGill

1-dimensional Hermite example

(1) Input x as a value

```
>> new=OrthogonalPolynomials(3);
>> C=HermiteCollocationMatrix(new,1)

C =

     1     1     0    -2
```

**Second line is the same as:*

```
C=new.HermiteCollocationMatrix(1)
```

(2) Input x as a vector

```
>> x=[1 2 3 4];
>> new=OrthogonalPolynomials(3);
>> C=HermiteCollocationMatrix(new,x)

C =

     1     1     0    -2
     1     2     3     2
     1     3     8    18
     1     4    15    52
```

McGill

Legendre and Laguerre collocation matrix

- Legendre recurrence relationship:

$$P_0(\xi) = 1 \quad P_1(\xi) = \xi$$

$$P_{n+1}(\xi) = \frac{2n+1}{n+1}\xi T_n(\xi) - \frac{n}{n+1}T_{n-1}(\xi)$$

- Laguerre recurrence relationship:

$$L_0^\alpha(\xi) = 1 \quad L_1^\alpha(\xi) = 1 + \alpha - \xi$$

$$L_{n+1}^\alpha(\xi) = \frac{(2n+1+\alpha-\xi)L_n^\alpha(\xi) - (n+\alpha)L_{n-1}^\alpha(\xi)}{n+1}$$

- Laguerre example (x as a vector):

```
>> x=[1 2 3 4];
>> new=OrthogonalPolynomials(3);
>> C=LaguerreCollocationMatrix(new,x,0

C =

    1.0000         0   -0.5000   -0.6667
    1.0000   -1.0000   -1.0000   -0.3333
    1.0000   -2.0000   -0.5000    1.0000
    1.0000   -3.0000    1.0000    2.3333
```

McGill

Roots for Hermite Polynomials

- **T-matrix for Hermite polynomials**

$$\boldsymbol{T} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 2 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & M \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

- **Example:**

```
new=OrthogonalPolynomials(3);
```

```
K>> roots=HermiteRoots(new)

roots =

   -2.3344
   -0.7420
    0.7420
    2.3344
```

McGill

Roots for Legendre and Laguerre Polynomials

- **T-matrix**

$$\boldsymbol{T} = \begin{bmatrix} 0 & \frac{1}{3} & 0 & \cdots & 0 & 0 \\ 1 & 0 & \frac{2}{5} & \cdots & 0 & 0 \\ 0 & \frac{2}{3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \frac{M}{2M+1} \\ 0 & 0 & 0 & \cdots & \frac{M}{2M-1} & 0 \end{bmatrix}$$

Legendre

$$\boldsymbol{T} = \begin{bmatrix} 1+\alpha & -(1+\alpha) & 0 & \cdots & 0 & 0 \\ -1 & 3+\alpha & -(2+\alpha) & \cdots & 0 & 0 \\ 0 & -2 & 5+\alpha & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2M-1+\alpha & -(M+\alpha) \\ 0 & 0 & 0 & \cdots & -M & 2M+1+\alpha \end{bmatrix}$$

Laguerre

- **Example:**

```
K>> roots=LegendreRoots(new)

roots =

   -0.8611
   -0.3400
    0.8611
    0.3400
```

```
>> roots=LaguerreRoots(new,0)

roots =

    0.3225
    1.7458
    4.5366
    9.3951
```

McGill

Total Order Index

- Tensor order:
  - We define the ordering by assuming, without loss of generality, that $0 \le \alpha_i \le M$ for $i=1,\cdots,d$ and denote the rank of a multi-index α by $|\alpha|$, where

$$|\boldsymbol{\alpha}| := \sum_{i=1}^{d} \alpha_i (M+1)^{(d-i)}$$

* M is the order and d is the dimension of polynomials

Example (M=2,d=2):

α_1	α_2	ϕ_α	$\|\boldsymbol{\alpha}\|$
0	0	ϕ_{00}	0
0	1	ϕ_{01}	1
0	2	ϕ_{02}	2
1	0	ϕ_{10}	3
1	1	ϕ_{11}	4
1	2	ϕ_{12}	5
2	0	ϕ_{20}	6
2	1	ϕ_{21}	7
2	2	ϕ_{22}	8

Tensor order

Lexicographic order

α_1	α_2	ϕ_α	$\|\boldsymbol{\alpha}\|$
0	0	ϕ_{00}	0
0	1	ϕ_{01}	1
0	2	ϕ_{02}	2
1	0	ϕ_{10}	3
1	1	ϕ_{11}	4
2	0	ϕ_{20}	6

Total order

McGill

Total Order Index Example

- API: object.RecursiveTotalOrderIndex(order,dimension)
- Example:

```
>> new=OrthogonalPolynomials(2);
>> H = RecursiveTotalOrderIndex(new,2,2)

H =

     0     0
     0     1
     0     2
     1     0
     1     1
     2     0
```

α_1	α_2	ϕ_α	$\|\alpha\|$
0	0	ϕ_{00}	0
0	1	ϕ_{01}	1
0	2	ϕ_{02}	2
1	0	ϕ_{10}	3
1	1	ϕ_{11}	4
2	0	ϕ_{20}	6

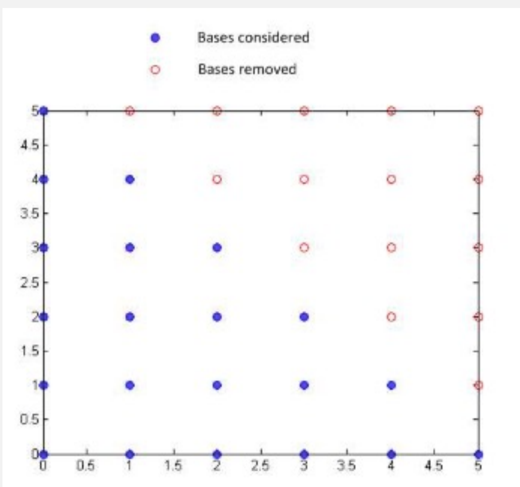
McGill

Hyperbolic Index

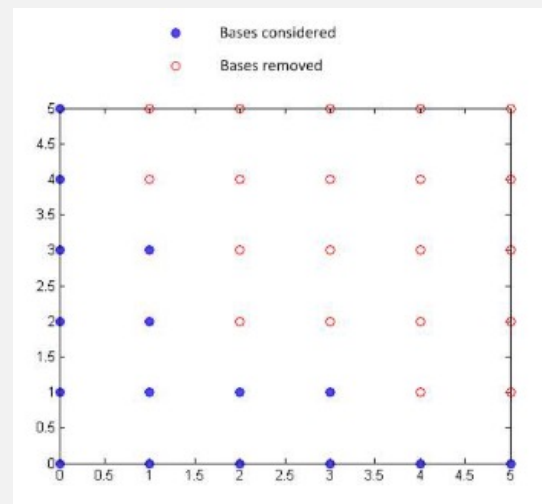
- Replace criterion in total order $\| \mathbf{d} \|_1 = d_1 + d_2 + \ldots. + d_n \leq m$

  with $\|d\|_u = \left(d_1^{\,u} + d_2^{\,u} + \ldots + d_n^{\,u}\right)^{1/u} \leq m$

- So, indices are limited by the hyperbola $\left(d_1^{\,u} + d_2^{\,u} + \ldots + d_n^{\,u}\right)^{1/u} = m$



Total order truncation



Hyperbolic truncation

McGill

Hyperbolic Index Example

- API: object.HyperbolicIndex(H,u,order)
  - H: the total order index with same dimension and order
  - u: the hyperbolic factor
  - order: order of the polynomials
- Example:

```
% create an object 'new' for the orthogonal polynomial with order 3
new=OrthogonalPolynomials(2);
% compute the total order index matrix of 'new' by specifying its
% dimension and order are both 2
H=RecursiveTotalOrderIndex(new,2,2);
% compute the hoperbolic index of 'new' by inputting 'H',u,and order
Hyper=HyperbolicIndex(new,H,0.8,2)
```

```
>> Hyper = HyperbolicIndex(new,H,0.8,2)

Hyper =

     0     0
     0     1
     0     2
     1     0
     2     0
```

McGill

Main Function: the total order

- API: object.TotalOrderMultiDCollocationMatrix(x,types)
  - x: the multi-dimensional sample point(s)
  - types: A cell array that contains string to describe the input types of each column of x. 'H' stands for Hermite polynomials, 'Leg' stands for Legendre polynomials, and 'Lag' stands for Laguerre polynomials.
- Example: Let’s assume we have the following set of sample points

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \\ 5 & 5 & 5 \end{bmatrix}$$

- Each column respectively corresponds to Hermite, Legendre, and Laguerre polynomial.

Example continued

- **Example codes:**

```
% specify the type of polynomials for each column
types={'H','Leg','Lag'};
% create the orthogonal polynomial object and name it 'new'
new = OrthogonalPolynomials(3);
% call the function to compute the collocation matrix
% with the sample and types
multiC=new.TotalOrderMultiDCollocationMatrix(x,types)
```

x=

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \\ 5 & 5 & 5 \end{bmatrix}$$

- **Results:**

```
multiC =

  Columns 1 through 12

    1.0000         0   -0.5000   -0.6667    1.0000         0   -0.5000    1.0000         0    1.0000    1.0000         0
    1.0000   -1.0000   -1.0000   -0.3333    2.0000   -2.0000   -2.0000    5.5000   -5.5000   17.0000    2.0000   -2.0000
    1.0000   -2.0000   -0.5000    1.0000    3.0000   -6.0000   -1.5000   13.0000  -26.0000   63.0000    3.0000   -6.0000
    1.0000   -3.0000    1.0000    2.3333    4.0000  -12.0000    4.0000   23.5000  -70.5000  154.0000    4.0000  -12.0000
    1.0000   -4.0000    3.5000    2.6667    5.0000  -20.0000   17.5000   37.0000 -148.0000  305.0000    5.0000  -20.0000

  Columns 13 through 20

   -0.5000    1.0000         0    1.0000         0         0         0   -2.0000
   -2.0000    4.0000   -4.0000   11.0000    3.0000   -3.0000    6.0000    2.0000
   -1.5000    9.0000  -18.0000   39.0000    8.0000  -16.0000   24.0000   18.0000
    4.0000   16.0000  -48.0000   94.0000   15.0000  -45.0000   60.0000   52.0000
   17.5000   25.0000 -100.0000  185.0000   24.0000  -96.0000  120.0000  110.0000
```

McGill

Main function: the hyperbolic index

- **Example codes:**

```
% specify the type of polynomials for each column
types={'H','Leg','Lag'};
% create the orthogonal polynomial object and name it 'new'
new = OrthogonalPolynomials(3);
% call the function to compute the collocation matrix
% with the sample and types
multiC=new.HyperbolicMultiDCollocationMatrix(x,types)
```

x= $\begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \\ 4 & 4 & 4 \\ 5 & 5 & 5 \end{bmatrix}$

- **Results**

```
multiC =

    1.0000         0   -0.5000    1.0000         0    1.0000    1.0000         0    1.0000         0
    1.0000   -1.0000   -1.0000    2.0000   -2.0000    5.5000    2.0000   -2.0000    4.0000    3.0000
    1.0000   -2.0000   -0.5000    3.0000   -6.0000   13.0000    3.0000   -6.0000    9.0000    8.0000
    1.0000   -3.0000    1.0000    4.0000  -12.0000   23.5000    4.0000  -12.0000   16.0000   15.0000
    1.0000   -4.0000    3.5000    5.0000  -20.0000   37.0000    5.0000  -20.0000   25.0000   24.0000
```

McGill

Thank you for your attention!