Forward uncertainty quantification (in high dimensions)

Wouter Edeling
CWI Research Semester Programme

October 6, 2024

Schedule

- Today: theory forward UQ (lecture notes are in the repo)
- Tomorrow: forward UQ with EasyVVUQ (coding session)

What is UQ?

What is UQ?

What is UQ?

- "Put error bars on numerical predictions"

What is UQ?

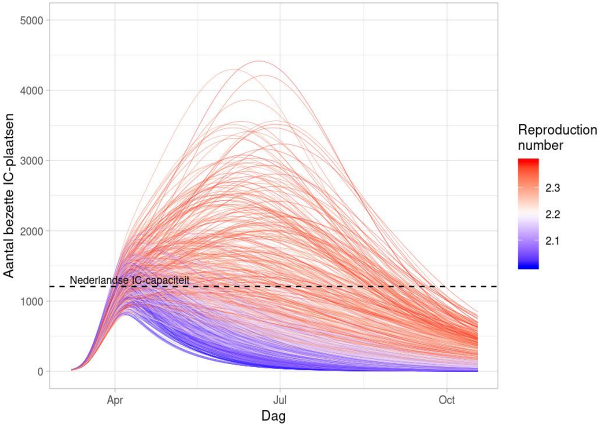
- “Put error bars on numerical predictions”
- If a model has uncertain input parameter(s), how uncertain is the model output (Quantity of Interest, QoI)?

What is UQ?

- "Put error bars on numerical predictions"
- If a model has uncertain input parameter(s), how uncertain is the model output (Quantity of Interest, QoI)?
- Quantitative assessment of uncertainties in model simulations and their impact on simulation output.

Why bother with UQ?

Example: uncertainty in epidemic modeling



No. of IC beds occupied by Covid-19 patients versus time [1]

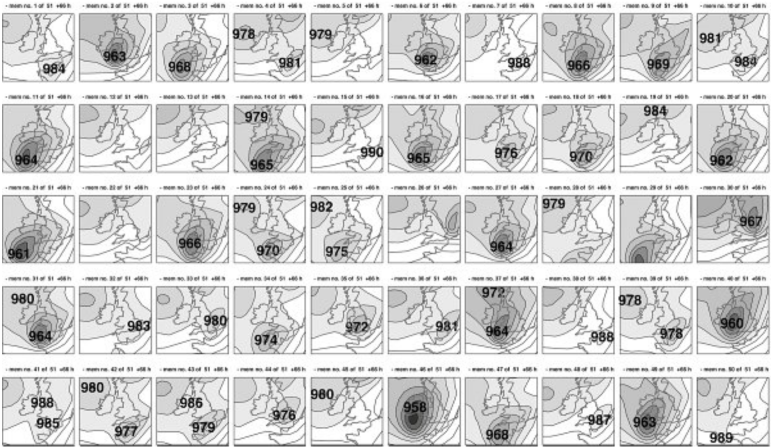
[1] Figure from a presentation to Dutch parliament by Jaap van Dissel, director of Center for Infectious Diseases of RIVM (25 March 2020).

Why bother with UQ?

Example: Michael Fish and the hurricane

Why bother with UQ?

- Afterwards: first study on uncertainty via ensemble forecast for Oct 16, 1987
- Perturb inputs, run model multiple time
- 30% of ensemble runs do predict a severe storm

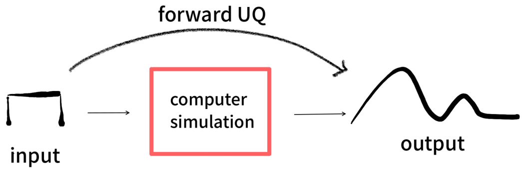


Why bother with UQ?

- Relevance for a wide range of applications
- Engineering, CFD, hydrology, reservoir modeling, climate science, biology, ...
- Major constraint: high computational cost of simulation

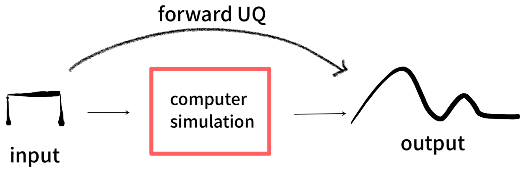
Some key UQ questions

- Given an input probability distribution, what is the output distribution? (forward UQ)
- What are the lowest output moments? (mean & variance)
- What are the tail probabilities? (extreme events)

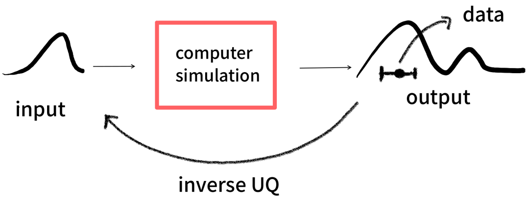


Some key UQ questions

- Given an input probability distribution, what is the output distribution? (forward UQ)
- What are the lowest output moments? (mean & variance)
- What are the tail probabilities? (extreme events)

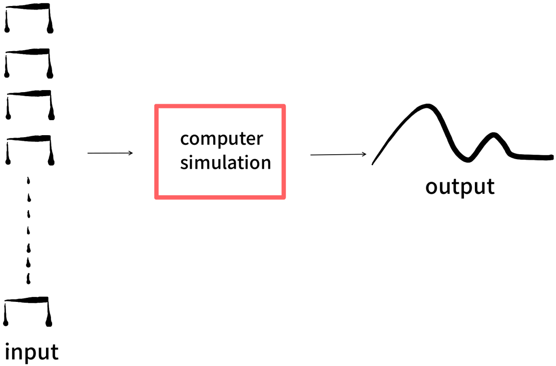


- Can we infer the input distribution of from output data of Y? (inverse UQ)



Some key UQ questions

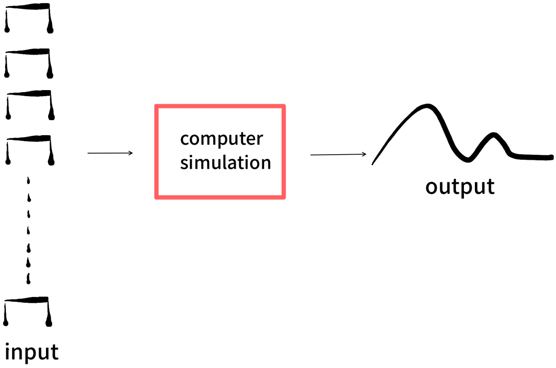
- Can I handle $\gg 1$ uncertain inputs? (high-dimensional UQ, today's topic)



- Which of these inputs are actually important? (sensitivity analysis)

Some key UQ questions

- Can I handle $\gg 1$ uncertain inputs? (high-dimensional UQ, today's topic)



- Which of these inputs are actually important? (sensitivity analysis)

(High-dimensional) forward UQ

Sampling

- To answer any UQ question: draw (random) samples from code
- Most well-know sampling method:

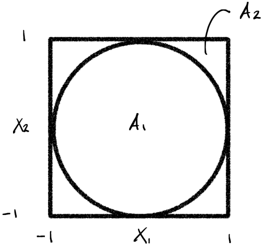
Sampling

- To answer any UQ question: draw (random) samples from code
- Most well-know sampling method: Monte Carlo sampling

- Let $x_1, x_2, \cdots, x_D$ be D (independent) input parameters,
- let $x_i \sim p(x_i)$, where p is a given probability density function,
- draw N samples from all $p(x_i)$: $x_i^{(k)}$, $k = 1, \cdots, N$,
- evaluate code $y^{(k)} = f(x_1^{(k)}, \cdots, x_D^{(k)})$,
- compute (stats of) output y, using $y^{(k)}$ samples.

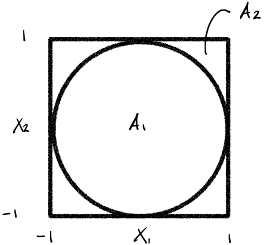
Sampling

Example: `scripts/examples/Monte_Carlo_example.ipynb`



Sampling

Example: `scripts/examples/Monte_Carlo_example.ipynb`



- Downside MC: slow $1/\sqrt{N}$ convergence,
- Upside MC: independent of D (number of inputs).

Sampling

Can we obtain a better convergence rate?

Sampling

Can we obtain a better convergence rate? Yes:

- Polynomials Chaos Expansions (PCE)
- Stochastic Collocation (SC)

PCE in 1D

PCE in 1D

- x is a scalar random input, with $x \sim p(x)$,
- $f(x)$ is an (expensive) Quantity of Interest (QoI),
- Polynomials Chaos Expansion of f:

$$\boxed{f(x) = \sum_{i=0}^{\infty} \hat{f}_i \phi_i(x)}$$

PCE in 1D

- x is a scalar random input, with $x \sim p(x)$,
- $f(x)$ is an (expensive) Quantity of Interest (QoI),
- Polynomials Chaos Expansion of f:

$$\boxed{f(x) = \sum_{i=0}^{\infty} \hat{f}_i \phi_i(x)}$$

- $\hat{f}_i$: unknown PCE coefficients.

PCE in 1D

- x is a scalar random input, with $x \sim p(x)$,
- $f(x)$ is an (expensive) Quantity of Interest (QoI),
- Polynomials Chaos Expansion of f:

$$\boxed{f(x) = \sum_{i=0}^{\infty} \hat{f}_i \phi_i(x)}$$

- $\hat{f}_i$: unknown PCE coefficients.
- ϕ_i: orthogonal polynomial basis in stochastic space x

PCE in 1D: orthogonal basis

- ϕ_i: orthogonal polynomial basis in stochastic space x
  - orthogonal w.r.t. what inner product?

PCE in 1D: orthogonal basis

- ϕ_i: orthogonal polynomial basis in stochastic space x
  - orthogonal w.r.t. what inner product?

$$\int \phi_i(x)\phi_j(x)p(x)\mathrm{d}x = \mathbb{E}\left[\phi_i\phi_j\right] = \delta_{ij}\gamma_i,$$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$\gamma_i = \mathbb{E}\left[\phi_i\phi_i\right]$$

basis vectors are orthonormal under the expected value

PCE in 1D: orthogonal basis

- ϕ_i: orthogonal polynomial basis in stochastic space x
  - orthogonal w.r.t. what inner product?

$$\int \phi_i(x)\phi_j(x)p(x)\mathrm{d}x = \mathbb{E}\left[\phi_i\phi_j\right] = \delta_{ij}\gamma_i,$$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$\gamma_i = \mathbb{E}\left[\phi_i\phi_i\right]$$

basis vectors are orthonormal under the expected value

PCE in 1D: orthogonal basis

- How to find orthogonal basis ϕ_i?
  - Compute yourself via Gram-Schmidt,
  - use available libraries (e.g. `chaospy`),
  - if $p(x)$ is a common distribution, look up corresponding ϕ_i

PCE in 1D: orthogonal basis

- How to find orthogonal basis ϕ_i?
  - Compute yourself via Gram-Schmidt,
  - use available libraries (e.g. `chaospy`),
  - if $p(x)$ is a common distribution, look up corresponding ϕ_i
- For instance:
  - $p(x)$ is Uniform: ϕ_i are Legendre polynomials.
  - $p(x)$ is Gaussian: ϕ_i are Hermite polynomials.

PCE in 1D: orthogonal basis

- Advantage orthogonal basis: mean & var are easy.
  - Mean:

$$\mathbb{E}\left[f\right] = \mathbb{E}\left[\sum_i \hat{f}_i \phi_i\right] = \sum_i \hat{f}_i \mathbb{E}\left[\phi_i\right] = \hat{f}_0$$

- Why?

PCE in 1D: orthogonal basis

- Advantage orthogonal basis: mean & var are easy.
  - Mean:

$$\mathbb{E}[f] = \mathbb{E}\left[\sum_i \hat{f}_i \phi_i\right] = \sum_i \hat{f}_i \mathbb{E}[\phi_i] = \hat{f}_0$$

- Why?
  - orthogonality & $\phi_0 = 1$:

$$\mathbb{E}[\phi_i] = \mathbb{E}[\underbrace{\phi_0}_{1} \phi_i] = \delta_{0i}\gamma_0 = \delta_{0i}$$

$$\gamma_0 := \mathbb{E}[\phi_0\phi_0] = \int 1 \; p(x)\mathrm{d}x = 1$$

PCE in 1D: orthogonal basis

- Advantage: moments are easily computed.
  - Variance:

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[f\right] := \mathbb{E}[(f - \mathbb{E}[f])^2] &= \mathbb{E}\left[\left(\sum_{i=0} \hat{f}_i \phi_i - \hat{f}_0\right)^2\right] \\ &= \mathbb{E}\left[\left(\sum_{i=1} \hat{f}_i \phi_i\right)^2\right] =?\end{aligned}$$

PCE in 1D: orthogonal basis

- Advantage: moments are easily computed.
  - Variance:

$$\mathbb{V}\mathrm{ar}\left[f\right] := \mathbb{E}[(f - \mathbb{E}[f])^2] = \mathbb{E}\left[\left(\sum_{i=0} \hat{f}_i\phi_i - \hat{f}_0\right)^2\right]$$
$$= \mathbb{E}\left[\left(\sum_{i=1} \hat{f}_i\phi_i\right)^2\right] =?$$

$$\mathbb{E}\left[\left(\sum_{i=1} \hat{f}_i\phi_i\right)^2\right] = \sum_{i=1}\sum_{j=1} \hat{f}_i\hat{f}_j\mathbb{E}\left[\phi_i\phi_j\right] = \sum_{i=1}\sum_{j=1} \hat{f}_i\hat{f}_j\delta_{ij}\gamma_j = \sum_{i=1} \hat{f}_i^2\gamma_i$$

PCE in 1D: orthogonal basis

- Recap, due to orthogonal basis:

$$\boxed{\mathbb{E}[f] = \hat{f}_0, \quad \mathbb{V}\mathrm{ar}[f] = \sum_{i=1} \hat{f}_i^2 \gamma_i}$$

mean and variance are simple functions of PCE coefficients.

PCE in 1D: orthogonal basis

- Recap, due to orthogonal basis:

$$\boxed{\mathbb{E}[f] = \hat{f}_0, \quad \mathbb{V}\mathrm{ar}[f] = \sum_{i=1} \hat{f}_i^2 \gamma_i}$$

mean and variance are simple functions of PCE coefficients.

- Remaining question:
  - How do we find PCE coefficients $\hat{f}_j$?

PCE in 1D: orthogonal basis

- Multiply f with ϕ_j and take expectation:

$$f = \sum_{i=0} \hat{f}_i \phi_i \Leftrightarrow$$

$$\mathbb{E}[f\phi_j] = \sum_{i=0} \hat{f}_i \, \mathbb{E}[\phi_i \phi_j] = \hat{f}_j \gamma_j$$

PCE in 1D: orthogonal basis

- Multiply f with ϕ_j and take expectation:

$$f = \sum_{i=0} \hat{f}_i \phi_i \Leftrightarrow$$

$$\mathbb{E}[f\phi_j] = \sum_{i=0} \hat{f}_i \, \mathbb{E}[\phi_i \phi_j] = \hat{f}_j \gamma_j$$

- And so:

$$\boxed{\hat{f}_j = \frac{\mathbb{E}[f\phi_j]}{\gamma_j}}$$

PCE in 1D: approximation

- How to compute $\mathbb{E}[f\phi_j]$?

PCE in 1D: approximation

- How to compute $\mathbb{E}[f\phi_j]$? quadrature:

$$\mathbb{E}[f\phi_j] = \int f(x)\phi_j(x)p(x)\mathrm{d}x \approx \sum_{k=1}^{Q} f(x_k)\phi_j(x_k)w_j$$

- This is where we sample our code ($f(x_k)$, $k = 1, \cdots, Q$)
- Use quadrature associated with $p(x)$!

PCE in 1D: approximation

- How to compute $\mathbb{E}[f\phi_j]$? quadrature:

$$\mathbb{E}[f\phi_j] = \int f(x)\phi_j(x)p(x)\mathrm{d}x \approx \sum_{k=1}^{Q} f(x_k)\phi_j(x_k)w_j$$

- This is where we sample our code ($f(x_k)$, $k = 1, \cdots, Q$)
- Use quadrature associated with $p(x)$!
- $\gamma_i = \mathbb{E}[\phi_i\phi_i]$ is computed exactly with quadrature.

PCE in 1D: approximation

- To recap:

$$f(x) = \sum_{i=0}^{\infty} \hat{f}_i \phi_i(x) \approx \sum_{i=0}^{N} \hat{f}_i \phi_i(x) \approx \sum_{i=0}^{N} \tilde{f}_i \phi_i(x)$$

PCE in 1D: approximation

- To recap:

$$f(x) = \sum_{i=0}^{\infty} \hat{f}_i \phi_i(x) \approx \sum_{i=0}^{N} \hat{f}_i \phi_i(x) \approx \sum_{i=0}^{N} \tilde{f}_i \phi_i(x)$$

where:

$$\tilde{f}_i = \frac{\sum_{k=1}^{Q} f(x_k)\phi_j(x_k)w_j}{\sum_{k=1}^{Q} \phi_j(x_k)\phi_j(x_k)w_j} \approx \frac{\mathbb{E}[f\phi_i]}{\mathbb{E}[\phi_i\phi_i]}$$

$$\mathbb{E}[f] = \hat{f}_0 \approx \tilde{f}_0, \quad \mathbb{V}\mathrm{ar}[f] = \sum_{i=1} \hat{f}_i^2 \gamma_i \approx \sum_{i=1} \tilde{f}_i^2 \gamma_i$$

Some approximation/sampling error is (almost) unavoidable.

Sampling

- How does the PCE sampling error compare against MC?
- Example: `scripts/examples/1D_PCE_example.ipynb`

MC vs PCE for $f(x) = \exp(2x + \sin(x))$, $x \sim \mathcal{U}[-1, 1]$

Sampling

- How does the PCE sampling error compare against MC?
- Example: `scripts/examples/1D_PCE_example.ipynb`

  MC vs PCE for $f(x) = \exp(2x + \sin(x))$, $x \sim \mathcal{U}[-1, 1]$

- PCE: rate of convergence depends on smoothness of $f(x)$
  the smoother $f(x)$, the smaller the error for given N.

PCE in higher dimensions

Multivariate input

- What if $\dim(\mathbf{x}) > 1$?

Multivariate input

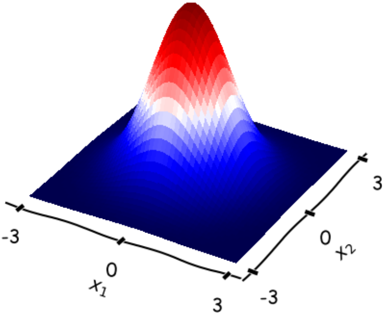
- What if $\dim(\mathbf{x}) > 1$?
- $\mathbf{x} = [x_1, \cdots, x_D]^T$ is now a multivariate random variable.
- Assume independence:

$$\mathbf{x} \sim p(\mathbf{x}) = \prod_{i=1}^{D} p(x_i)$$

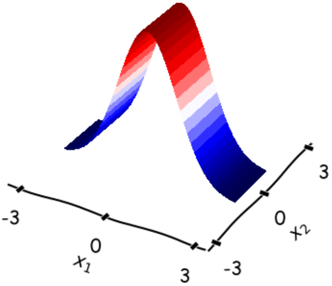
joint pdf is factorized as product 1D pdf's.

Multivariate input

- Joint pdf examples:



(a) $x_1, x_2 \sim \mathcal{N}(0,1)$



(b) $x_1 \sim \mathcal{N}(0,1)$, $x_2 \sim \mathcal{U}(-1,1)$

Multivariate basis

- PCE basis in 1D:

$$\phi_i(x), \quad i = 1, \cdots, N$$

Multivariate basis

- PCE basis in 1D:

$$\phi_i(x), \quad i = 1, \cdots, N$$

- Multivariate PCE basis:

$$\Phi_{\mathbf{i}}, \quad \mathbf{i} \in \Lambda$$

- What is Φ?
- What is $\mathbf{i}$?
- What is Λ?

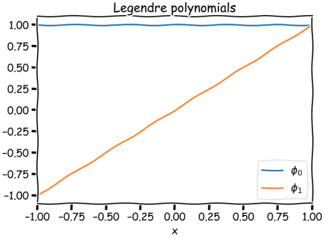
Multivariate basis in D dimensions

- We count over multi indices:

$$\boxed{\mathbf{i} := (i_1, i_2, \cdots, i_D)}$$

Each i_j selects 1D basis ϕ_{i_j} for j-th input x_j.

e.g.: $\mathbf{i} = (0, 1)$ selects $\phi_0(x_1)$ and $\phi_1(x_2)$ if $D = 2$.

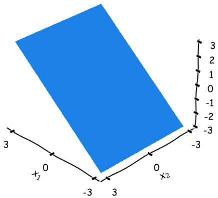


Multivariate basis in D dimensions

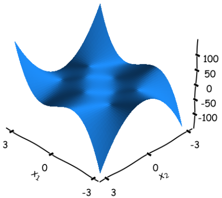
- 1 Basis function:

$$\Phi_{\mathbf{i}} = \phi_{i_1}(x_1)\phi_{i_2}(x_2)\cdots\phi_{i_D}(x_D) = \prod_{j=1}^{D} \phi_{i_j}(x_j)$$

product of the 1D bases selected by $\mathbf{i} \in \Lambda$.



(a) $\phi_1(x_1)\phi_0(x_2)$



(b) $\phi_3(x_1)\phi_2(x_2)$

Multivariate truncation

- As in 1D, we must truncate the PCE expansion:
  $\rightarrow \mathbf{i} \in \Lambda$: the D-dimensional equivalent of N: $\sum_{i=0}^{N}$

Multivariate truncation

- As in 1D, we must truncate the PCE expansion:
  $\rightarrow \mathbf{i} \in \Lambda$: the *$D$-dimensional equivalent of N*: $\sum_{i=0}^{N}$

- Multi-dimensional UQ:

Which $\mathbf{i}$ do I admit into Λ?

Multivariate truncation

- As in 1D, we must truncate the PCE expansion:
  $\rightarrow \mathbf{i} \in \Lambda$: the D-dimensional equivalent of N: $\sum_{i=0}^{N}$

- Multi-dimensional UQ:

Which $\mathbf{i}$ do I admit into Λ?

- High-dimensional UQ ($D \gg 5$):

Which $\underline{\text{important}}$ $\mathbf{i}$ do I admit into Λ?

Multivariate PCE

- Multivariate PCE expansion:

$$\boxed{f(\mathbf{x}) = \sum_{\mathbf{i} \in \Lambda} \hat{f}_{\mathbf{i}} \Phi_{\mathbf{i}}(\mathbf{x})}$$

with again:

$$\hat{f}_{\mathbf{i}} := \frac{\mathbb{E}\left[f \Phi_{\mathbf{i}}\right]}{\gamma_{\mathbf{i}}}, \quad \gamma_{\mathbf{i}} := \mathbb{E}\left[\Phi_{\mathbf{i}} \Phi_{\mathbf{i}}\right]$$

Multivariate PCE

- except now expectations are multi-dimensional, e.g. for $D = 2$:

$$\mathbb{E}\left[f\Phi_{\mathbf{i}}\right] = \int\int f(x_1, x_2)\ \phi_{i_1}(x_1)\phi_{i_2}(x_2)\ p(x_1)p(x_2)\ \mathrm{d}x_1\mathrm{d}x_2$$

$$\approx \sum_k \sum_l f\left(x_{1,k},\ x_{2,l}\right)\ \phi_{i_1}\left(x_{1,k}\right)\phi_{i_2}\left(x_{2,l}\right)\ w_k w_l$$

Multivariate PCE

- except now expectations are multi-dimensional, e.g. for $D = 2$:

$$\mathbb{E}\left[f \Phi_{\mathbf{i}}\right] = \int\int f(x_1, x_2)\ \phi_{i_1}(x_1)\phi_{i_2}(x_2)\ p(x_1)p(x_2)\ \mathrm{d}x_1\mathrm{d}x_2$$

$$\approx \sum_k \sum_l f\left(x_{1,k},\ x_{2,l}\right)\ \phi_{i_1}\left(x_{1,k}\right)\phi_{i_2}\left(x_{2,l}\right)\ w_k w_l$$

$$\begin{aligned}
\gamma_{\mathbf{i}} = \mathbb{E}\left[\Phi_{\mathbf{i}}\Phi_{\mathbf{i}}\right] &= \int\int \phi_{i_1}(x_1)\phi_{i_2}(x_2)\ \phi_{i_1}(x_1)\phi_{i_2}(x_2)\ p(x_1)p(x_2)\ \mathrm{d}x_1\mathrm{d}x_2 \\
&= \int \phi_{i_1}(x_1)\phi_{i_1}(x_1)p(x_1)\mathrm{d}x_1 \int \phi_{i_2}(x_2)\phi_{i_2}(x_2)p(x_2)\mathrm{d}x_2 \\
&=: \gamma_{i_1}\gamma_{i_2}
\end{aligned}$$

Multi indices

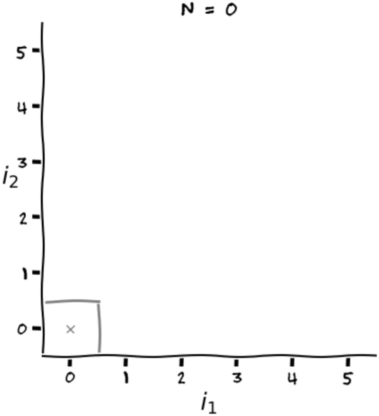
- Multi-dimensional UQ ($D \leq 5$): two common choices for Λ:

$$\Lambda = \{\mathbf{i} \mid |\mathbf{i}| \leq N\} \quad \text{or} \quad \Lambda = \{\mathbf{i} \mid \max(\mathbf{i}) \leq N\}$$

  Here: $|\mathbf{i}| = i_1 + i_2 + \cdots + i_D$

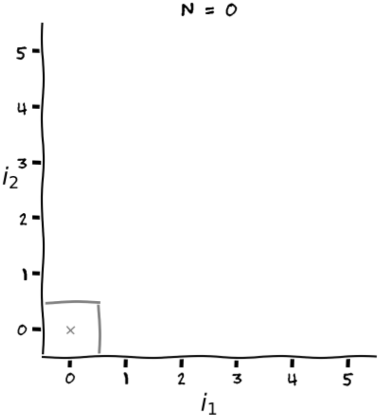
Multi indices in 2D: $\mathbf{i} = (i_1, i_2)$

$$\Lambda = \{\mathbf{i} \mid |\mathbf{i}| \leq N\}$$



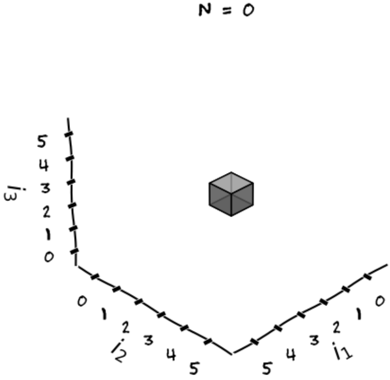
Multi indices in 2D: $\mathbf{i} = (i_1, i_2)$

$$\Lambda = \{\mathbf{i} \mid \max(\mathbf{i}) \leq N\}$$



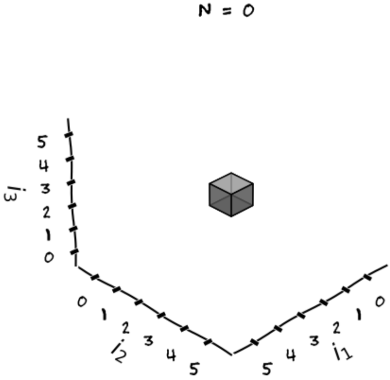
Multi indices in 3D: $\mathbf{i} = (i_1, i_2, i_3)$

$$\Lambda = \{\mathbf{i} \mid |\mathbf{i}| \leq N\}$$



Multi indices in 3D: $\mathbf{i} = (i_1, i_2, i_3)$

$$\Lambda = \{\mathbf{i} \mid \max(\mathbf{i}) \leq N\}$$



Multivariate moment estimation

- Once all $\hat{f}_{\mathbf{i}} \in \Lambda$ are computed:

$$\mathbb{E}[f] = \hat{f}_{\mathbf{0}}, \quad \mathbb{V}\mathrm{ar}[f] = \sum_{\substack{\mathbf{i} \in \Lambda \\ \mathbf{i} \neq \mathbf{0}}} \hat{f}_{\mathbf{i}}^2 \gamma_{\mathbf{i}}$$

- Or, use as surrogate model by sampling:

$$f(\mathbf{x}) = \sum_{\mathbf{i} \in \Lambda} \hat{f}_{\mathbf{i}} \Phi_{\mathbf{i}}(\mathbf{x})$$

Example: `scripts/examples/2D_PCE_example.ipynb`

Sampling

Remember: Can we obtain a better convergence rate? Yes:

- ✓ Polynomials Chaos Expansions (PCE)
- ▶ Stochastic Collocation (SC)

SC in 1D

SC in a nutshell

- PDE/model: e.g. $\mathcal{L}(u; \mathbf{x}) = \mathcal{F}(\mathbf{x})$, $\mathbf{x} \sim \prod_i p(x_i)$
- Define QoI: $f(u; \mathbf{x})$, e.g. $f = u$
- Select set of nodes: $\Theta_M := \{\mathbf{x}_j\}_{j=1}^N$
- Solutions: Solve PDE $\forall \mathbf{x}_j \in \Theta_N$, solutions $f_j := f(u; \mathbf{x}_j)$
- Construct approximation (surrogate) $\tilde{f} \approx f$, using $\{f_j\}_{j=1}^N$

standard SC in 1D

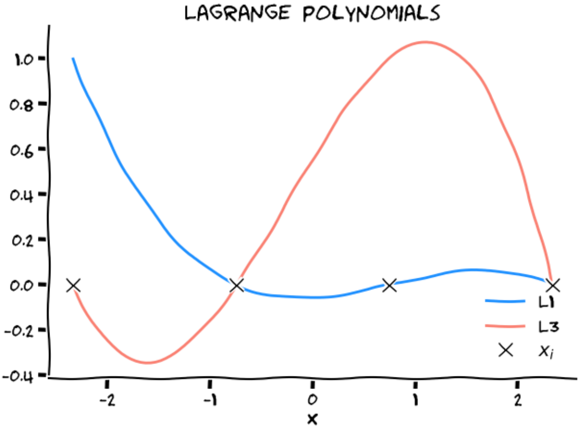
Approximation by interpolation operator I:

$$f(x) \approx \tilde{f}(x) = \mathrm{I}f(x) = \sum_{i=1}^{N} f(x_i)a_i(x).$$

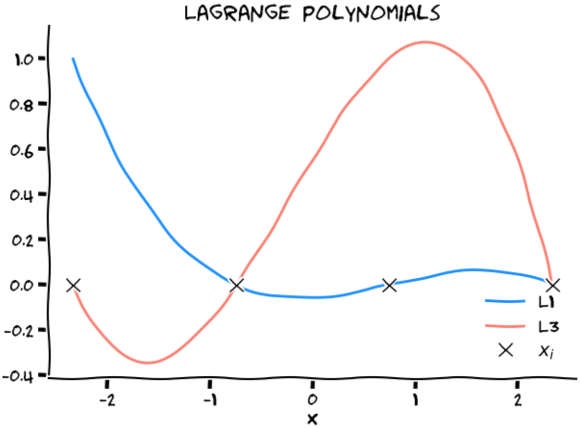
Common basis functions: Lagrange polynomials:

$$a_i^{(l)}(x) = \prod_{\substack{1 \leq j \leq m \\ j \neq i}} \frac{x - x_j}{x_i - x_j}$$

standard SC in 1D



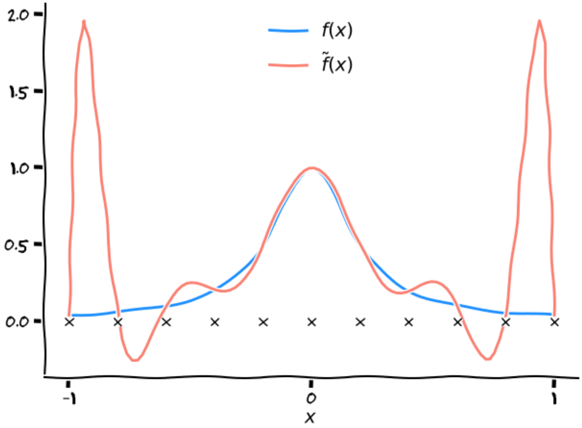
standard SC in 1D



How to choose the $\Theta_m := \{x_i\}_{i=1}^N$?

standard SC in 1D

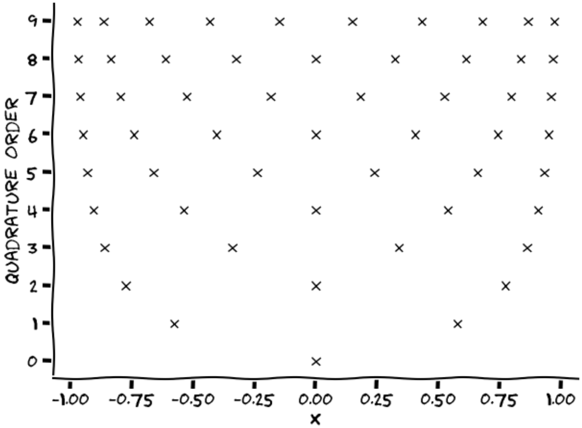
Uniform $\Theta_m := \{x_i\}_{i=1}^{N}$: Runge phenomenon



standard SC in 1D

Better choice:

- Same as in 1D PCE: (non-uniform) quadrature points



standard SC in 1D

- 1D SC
  - Evaluate code $\forall x_i$
  - Expansion coefficients: $f(x_i)$
  - Lagrange polynomials

$$f(x) \approx \tilde{f}(x) = \mathrm{I}f = \sum_{i=1}^{N} f(x_i) a_i(x)$$

- Compute mean & var of $\tilde{f}$
  e.g. $\mathbb{E}[\tilde{f}] = \sum_i f(x_i) w_i$

[2] Eldred, Michael, and John Burkardt. "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification." 47th AIAA aerospace sciences meeting. 2009.

standard SC in 1D

- 1D SC
  - Evaluate code $\forall x_i$
  - Expansion coefficients: $f(x_i)$
  - Lagrange polynomials

$$f(x) \approx \tilde{f}(x) = \mathrm{I}f = \sum_{i=1}^{N} f(x_i) a_i(x)$$

- Compute mean & var of $\tilde{f}$
  e.g. $\mathbb{E}[\tilde{f}] = \sum_i f(x_i) w_i$

- 1D PCE
  - Evaluate code $\forall x_i$
  - Expansion coefficients: $\hat{f}_i$
  - Orthogonal polynomials

$$f(x) \approx \tilde{f}(x) = \sum_{i=0}^{N} \hat{f}_i \phi_i(x)$$

- Compute mean & var of $\tilde{f}$
  e.g. $\mathbb{E}[\tilde{f}] = \hat{f}_0$

[2] Eldred, Michael, and John Burkardt. "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification." 47th AIAA aerospace sciences meeting. 2009.

standard SC in 1D

- 1D SC
  - Evaluate code $\forall x_i$
  - Expansion coefficients: $f(x_i)$
  - Lagrange polynomials

$$f(x) \approx \tilde{f}(x) = \mathrm{I}f = \sum_{i=1}^{N} f(x_i) a_i(x)$$

- Compute mean & var of $\tilde{f}$

  e.g. $\mathbb{E}[\tilde{f}] = \sum_i f(x_i) w_i$

- 1D PCE
  - Evaluate code $\forall x_i$
  - Expansion coefficients: $\hat{f}_i$
  - Orthogonal polynomials

$$f(x) \approx \tilde{f}(x) = \sum_{i=0}^{N} \hat{f}_i \phi_i(x)$$

- Compute mean & var of $\tilde{f}$

  e.g. $\mathbb{E}[\tilde{f}] = \hat{f}_0$

Often little difference in performance[2].

[2] Eldred, Michael, and John Burkardt. "Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification." 47th AIAA aerospace sciences meeting. 2009.

SC in higher dimensions

Multivariate SC

- What if $\dim(\mathbf{x}) > 1$?

Multivariate SC

- What if $\dim(\mathbf{x}) > 1$? Again:
- $\mathbf{x} = [x_1, \cdots, x_D]^T$ is now a multivariate random variable.
- Assume independence:

$$\mathbf{x} \sim p(\mathbf{x}) = \prod_{i=1}^{D} p(x_i)$$

Multivariate SC

- What if $\dim(\mathbf{x}) > 1$? Again:
- $\mathbf{x} = [x_1, \cdots, x_D]^T$ is now a multivariate random variable.
- Assume independence:

$$\mathbf{x} \sim p(\mathbf{x}) = \prod_{i=1}^{D} p(x_i)$$

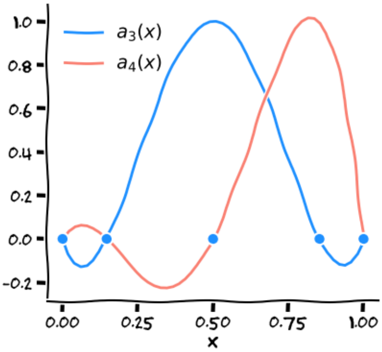
- 1 Basis function:

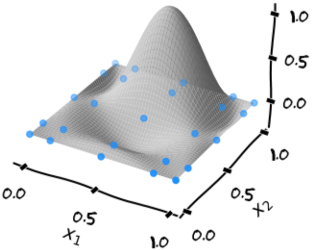
$$a_{\mathbf{j}} = a_{j_1}(x_1) \otimes a_{j_2}(x_2) \otimes \cdots \otimes a_{j_D}(x_D)$$

product of the 1D bases for each combination of quad points.

Multivariate SC

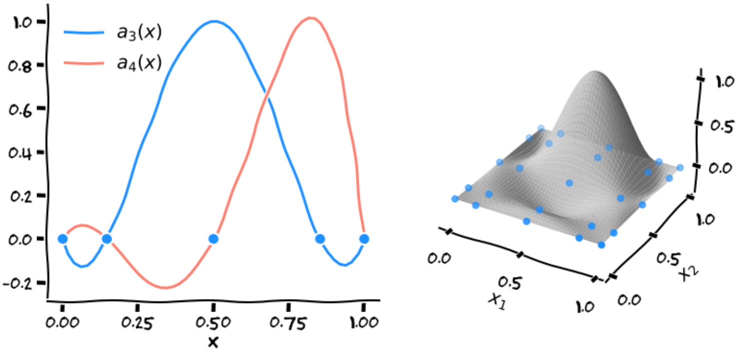
- Example 2D $a_{\mathbf{j}} = a_{j_1}(x_1) \otimes a_{j_2}(x_2)$ for a single 2D quad point (x_{j_1}, x_{j_2})





Multivariate SC

- Example 2D $a_{\mathbf{j}} = a_{j_1}(x_1) \otimes a_{j_2}(x_2)$ for a single 2D quad point (x_{j_1}, x_{j_2})



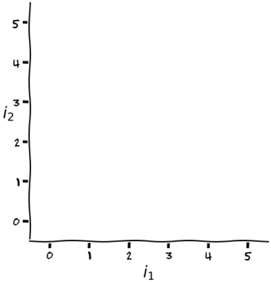
- Note: each $a_{\mathbf{j}}$ has the same polynomial order
  $\rightarrow$ **j** indexes quad points, not polynomial order

Multivariate SC

- Multivariate SC expansion:

$$f(\mathbf{x}) \approx \mathrm{I}f(\mathbf{x}) = \sum_{\mathbf{j}} f(\mathbf{x_j}) a_{\mathbf{j}}(\mathbf{x})$$
$$= \sum_{j_1=1}^{N_1} \cdots \sum_{j_D=1}^{N_D} f(x_{j_1}, \cdots, x_{j_D}) a_{j_1}(x_1) \cdots a_{j_D}(x_D)$$

- If $\mathbf{i}$ does index polynomial order, what is the SC multi-index set $\mathbf{i} \in \Lambda$?

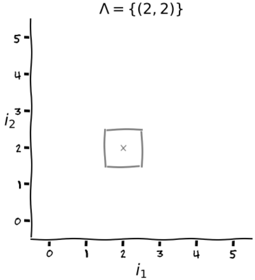


Multivariate SC

- Multivariate SC expansion:

$$f(\mathbf{x}) \approx \mathrm{I}f(\mathbf{x}) = \sum_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}) a_{\mathbf{j}}(\mathbf{x})$$
$$= \sum_{j_1=1}^{N_1} \cdots \sum_{j_D=1}^{N_D} f(x_{j_1}, \cdots, x_{j_D}) a_{j_1}(x_1) \cdots a_{j_D}(x_D)$$

- If $\mathbf{i}$ does index polynomial order, what is the SC multi-index set $\mathbf{i} \in \Lambda$?



Multivariate SC

- Rewrite SC expansion as:

$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f(\mathbf{x}) = \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{\mathbf{j}} f\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right) a_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x}),$$

  - with user-defined $\mathbf{l} = (l_1, \cdots, l_D)$
  - l_j determines the order of 1D quad rule used for x_j: $\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})} = x_{j_1}^{(l_1)} \otimes \cdots \otimes x_{j_D}^{(l_D)}$
  - $c_{\mathbf{l}} = 1$ (in this case).

Multivariate SC

- Rewrite SC expansion as:

$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f(\mathbf{x}) = \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{\mathbf{j}} f\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right) a_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x}),$$

  - with user-defined $\mathbf{l} = (l_1, \cdots, l_D)$
  - l_j determines the order of 1D quad rule used for x_j: $\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})} = x_{j_1}^{(l_1)} \otimes \cdots \otimes x_{j_D}^{(l_D)}$
  - $c_{\mathbf{l}} = 1$ (in this case).

- Why this is useful will be clear later.
- First, let's examine the cost of sampling $f(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})})$.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1 parameter, simulation time $=$ 10 seconds.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1 parameter, simulation time $=$ 10 seconds.

2 parameters, simulation time $\approx$ 1.5 minutes.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1 parameter, simulation time $= 10$ seconds.

2 parameters, simulation time ≈ 1.5 minutes.

3 parameters, simulation time ≈ 15 minutes.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1 parameter, simulation time $= 10$ seconds.

2 parameters, simulation time ≈ 1.5 minutes.

3 parameters, simulation time ≈ 15 minutes.

4 parameters, simulation time ≈ 3 hours.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1 parameter, simulation time $= 10$ seconds.

2 parameters, simulation time ≈ 1.5 minutes.

3 parameters, simulation time ≈ 15 minutes.

4 parameters, simulation time ≈ 3 hours.

5 parameters, simulation time ≈ 1.1 days.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1. parameter, simulation time $= 10$ seconds.
2. parameters, simulation time ≈ 1.5 minutes.
3. parameters, simulation time ≈ 15 minutes.
4. parameters, simulation time ≈ 3 hours.
5. parameters, simulation time ≈ 1.1 days.
6. parameters, simulation time ≈ 1.5 weeks.

Curse of dimensionality

Example:

- You have a fast code that takes 1 second to complete.
- You select $N_{j_i} = 10$ quad points per input parameter.

1 parameter, simulation time $= 10$ seconds.

2 parameters, simulation time ≈ 1.5 minutes.

3 parameters, simulation time ≈ 15 minutes.

4 parameters, simulation time ≈ 3 hours.

5 parameters, simulation time ≈ 1.1 days.

6 parameters, simulation time ≈ 1.5 weeks.

...

20 parameters, simulation time $\approx 3.17 \cdot 10^{12}$ years, 226 times the age of the universe.

Curse of dimensionality

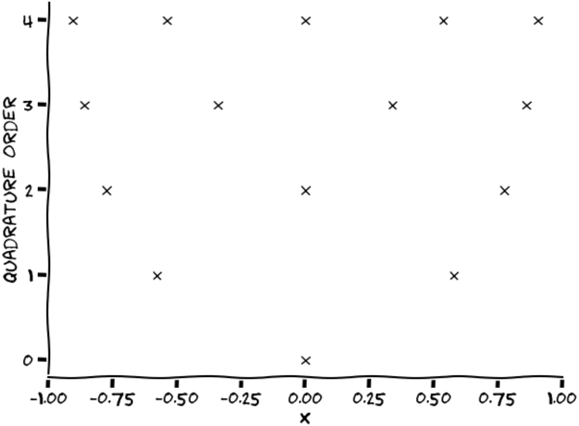
- In general: $N = N_{j_1} \times N_{j_2} \times \cdots \times N_{j_D}$
- Curse of dimensionality: exponential increase with D

No supercomputer can beat this, need smarter algorithms.

Efficient 1D quad rules

- First "trick":

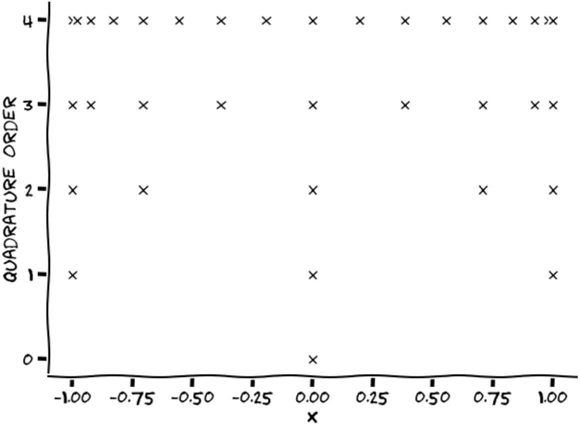
  → what is a problem with this 1D quad rule?



Efficient 1D quad rules

- First "trick":

  → nested 1D quad rule: reuse samples when refining



Nested 1D rule

- Well-known nested rule: Clenshaw-Curtis quadrature.
- Different "levels" (quad orders) over $x \in [0, 1]$

- Level $l = 1$: $x_i^{(1)} \in \{0.5\}$,
- Level $l = 2$: $x_i^{(2)} \in \{0.0,\ 0.5,\ 1.0\}$,
- Level $l = 3$: $x_i^{(3)} \in \{0.0,\ 0.146,\ 0.5,\ 0.854,\ 1.0\}$.

Nested 1D rule

- Well-known nested rule: Clenshaw-Curtis quadrature.
- Different "levels" (quad orders) over $x \in [0, 1]$
  - Level $l = 1$: $x_i^{(1)} \in \{0.5\}$,
  - Level $l = 2$: $x_i^{(2)} \in \{0.0,\ 0.5,\ 1.0\}$,
  - Level $l = 3$: $x_i^{(3)} \in \{0.0,\ 0.146,\ 0.5,\ 0.854,\ 1.0\}$.

Exponential increase in number of points per level;

$$N_l = \begin{cases} 2^{l-1} + 1 & l > 1 \\ 1 & l = 1 \end{cases}$$

but useful for refinement.

Nested 1D rule

- Well-known nested rule: Clenshaw-Curtis quadrature.
- Different "levels" (quad orders) over $x \in [0, 1]$
  - Level $l = 1$: $x_i^{(1)} \in \{0.5\}$,
  - Level $l = 2$: $x_i^{(2)} \in \{0.0,\ 0.5,\ 1.0\}$,
  - Level $l = 3$: $x_i^{(3)} \in \{0.0,\ 0.146,\ 0.5,\ 0.854,\ 1.0\}$.

Exponential increase in number of points per level;

$$N_l = \begin{cases} 2^{l-1} + 1 & l > 1 \\ 1 & l = 1 \end{cases}$$

but useful for refinement.

Weights are such that $\sum_i f(x_i) w_i \approx \int f(x) p(x) dx = \mathbb{E}[f]$.

Sparse grids

- What else can we do to postpone the curse?
- Multi-dimensional UQ:

Which $\mathbf{i}$ do I admit into Λ?

PCE: $|\mathbf{i}| \leq N$, SC: single user-defined $\mathbf{i}$.

Sparse grids

- What else can we do to postpone the curse?
- Multi-dimensional UQ:

$$\boxed{\text{Which } \mathbf{i} \text{ do I admit into } \Lambda?}$$

PCE: $|\mathbf{i}| \leq N$, SC: single user-defined $\mathbf{i}$.

- High-dimensional UQ ($D \gg 5$):

$$\boxed{\text{Which } \underline{\text{important}} \ \mathbf{i} \text{ do I admit into } \Lambda?}$$

(Dimension-adaptive) sparse-grid stochastic collocation

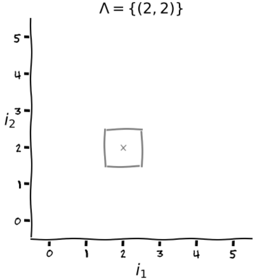
Sparse grids

- Basic idea (dimension-adaptive) sparse grids:

  - Be selective: only add "important" $\mathbf{l}$ to Λ
  - Do so iteratively: refine sampling plan in steps,
    $\rightarrow$ start from e.g. $\Lambda = \{(0, 0, \cdots, 0)\}$.

Sparse grids: iterative sampling

- Do not use 1 user-defined Λ:



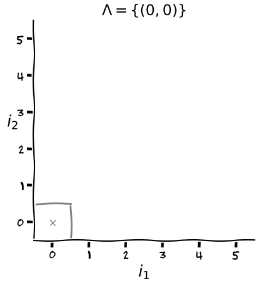
$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f(\mathbf{x}) = \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}}\, \mathrm{I}^{(l_1)} \otimes \mathrm{I}^{(l_2)} f(\mathbf{x})$$

$$= \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{j_1=1}^{N_{l_1}} \sum_{j_2=1}^{N_{l_2}} f\left(x_{j_1}^{(l_1)}, x_{j_2}^{(l_2)}\right) a_{j_1}^{(l_1)}(x_1) \otimes a_{j_2}^{(l_2)}(x_2)$$

$$c_{\mathbf{l}} = 1.$$

Sparse grids: iterative sampling

- But from start here, and then refine:



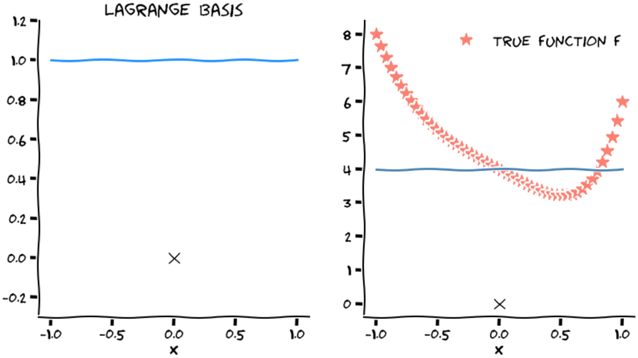
$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f(\mathbf{x}) = \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}}\, \mathrm{I}^{(l_1)} \otimes \mathrm{I}^{(l_2)} f(\mathbf{x})$$

$$= \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{j_1=1}^{N_{l_1}} \sum_{j_2=1}^{N_{l_2}} f\left(x_{j_1}^{(l_1)}, x_{j_2}^{(l_2)}\right) a_{j_1}^{(l_1)}(x_1) \otimes a_{j_2}^{(l_2)}(x_2)$$

$$c_{\mathbf{l}} = 1.$$

Sparse grids: iterative sampling

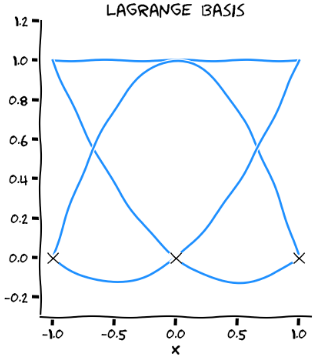
- Refinement in 1D: combining multiple surrogates

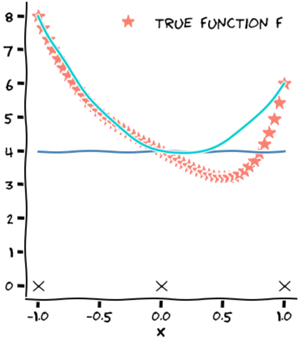


$$\mathrm{I}^{(0)} f(x)$$

Sparse grids: iterative sampling

- Refinement in 1D: combining multiple surrogates

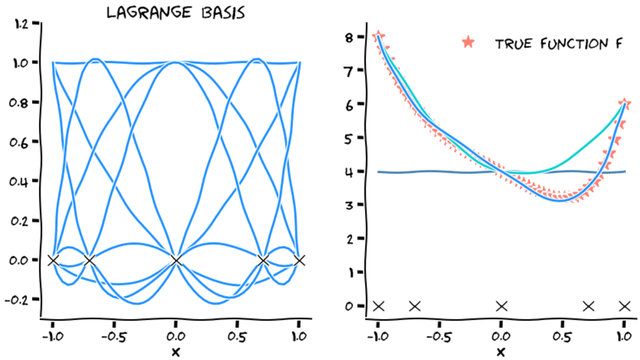




$$\mathrm{I}^{(1)} f(x)$$

Sparse grids: iterative sampling

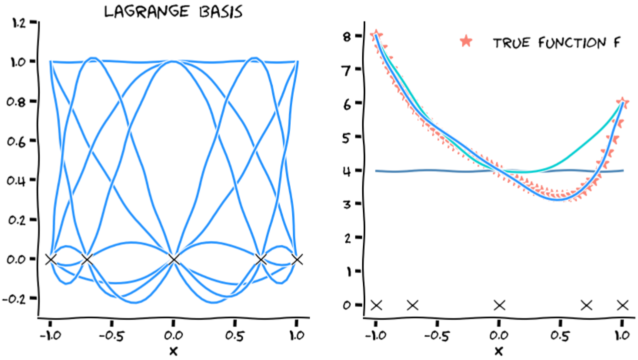
- Refinement in 1D: combining multiple surrogates



$$\mathrm{I}^{(2)} f(x) = ?$$

Sparse grids: iterative sampling

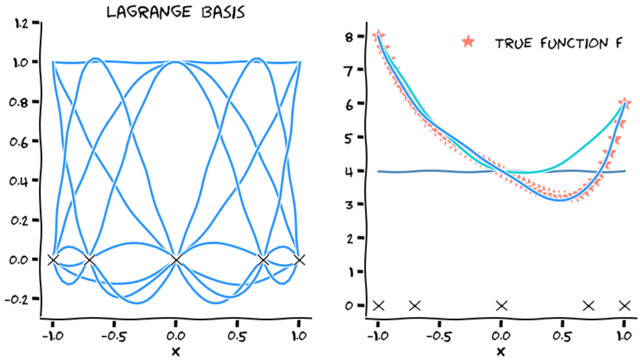
- Refinement in 1D: combining multiple surrogates



$$\mathrm{I}^{(2)} f(x) \neq \sum_{i=0}^{2} \mathrm{I}^{(i)} f$$

Sparse grids: iterative sampling

- Refinement in 1D: combining multiple surrogates



$$\mathrm{I}^{(2)}f(x) = \left(\mathrm{I}^{(2)}f(x) - \mathrm{I}^{(1)}f(x)\right) + \left(\mathrm{I}^{(1)}f(x) - \mathrm{I}^{(0)}f(x)\right) + \mathrm{I}^{(0)}f(x)$$

SC sparse grids: refinement in 1D

- Define 1D difference formulas:

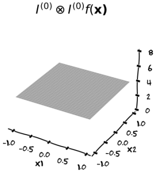
$$\Delta^{(l)} f := (\mathrm{I}^{(l)} - \mathrm{I}^{(l-1)}) f \quad \text{where} \quad \mathrm{I}^{(-1)} f := 0.$$

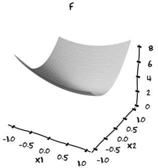
- Build (1D) $\tilde{f}$ as a telescopic sum of $\Delta \tilde{f}$'s

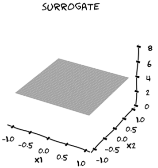
$$\begin{aligned}
\tilde{f}^{(L)} =& \mathrm{I}^{(0)} f \\
& \mathrm{I}^{(1)} f - \mathrm{I}^{(0)} f + \\
& \mathrm{I}^{(2)} f - \mathrm{I}^{(1)} f + \\
& \cdots + \\
& \mathrm{I}^{(L)} f - \mathrm{I}^{(L-1)} f = \sum_{l=0}^{L} \Delta^{(l)} f
\end{aligned}$$

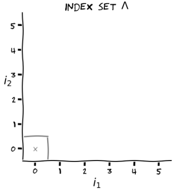
where each $\Delta^{(l)} f$ refines the expansion of previous level.

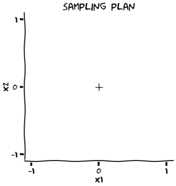
SC sparse grids: refinement in 2D





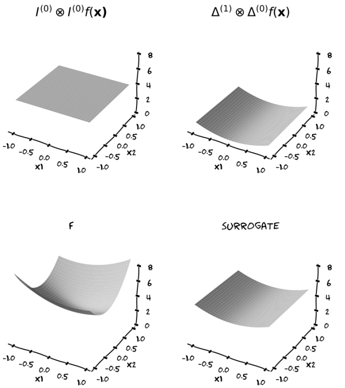


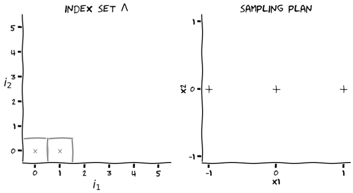




$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f = \mathrm{I}^{(0)} \otimes \mathrm{I}^{(0)} f(\mathbf{x})$$

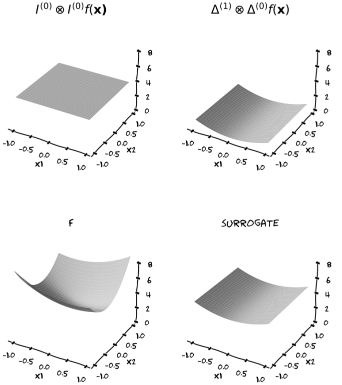
SC sparse grids: refinement in 2D

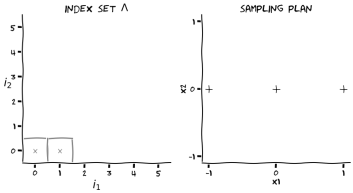




$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f \neq \mathrm{I}^{(0,0)} f(\mathbf{x}) + \mathrm{I}^{(1,0)} f(\mathbf{x})$$

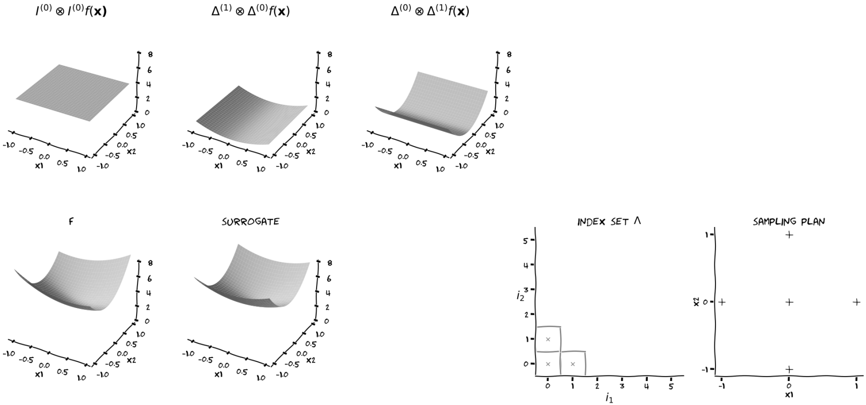
SC sparse grids: refinement in 2D





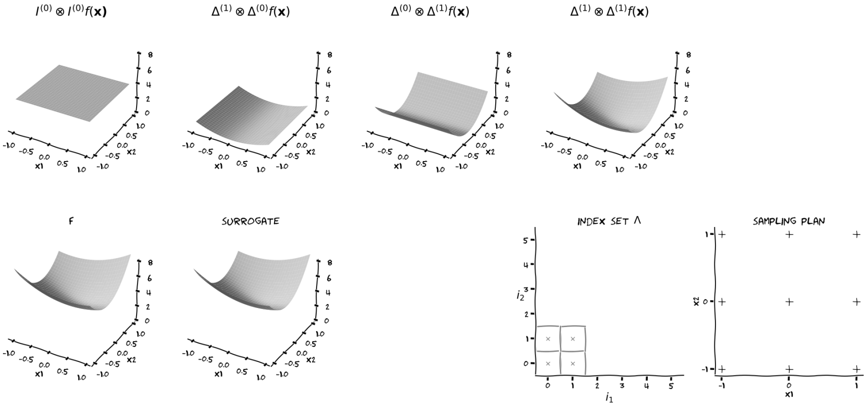
$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f = \mathrm{I}^{(0,0)} f(\mathbf{x}) + \Delta^{(1)} \otimes \Delta^{(0)} f$$

SC sparse grids: refinement in 2D



$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f = \mathrm{I}^{(0,0)} f(\mathbf{x}) + \Delta^{(1)} \otimes \Delta^{(0)} f + \Delta^{(0)} \otimes \Delta^{(1)} f$$

SC sparse grids: refinement in ND



$$f(\mathbf{x}) \approx \mathrm{I}^{(\Lambda)} f = \mathrm{I}^{(0,0)} f(\mathbf{x}) + \Delta^{(1)} \otimes \Delta^{(0)} f + \Delta^{(0)} \otimes \Delta^{(1)} f + \Delta^{(1)} \otimes \Delta^{(1)} f$$

SC sparse grids: refinement in ND

- We get a "multi dimensional telescopic sum":

$$\boxed{\mathrm{I}^{(\Lambda)} f = \sum_{\mathbf{l} \in \Lambda} \Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f(\mathbf{x})}$$

- Clearly displays concept of successive refinements.

Note:

$$\Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f := \left(\mathrm{I}^{(l_1)} - \mathrm{I}^{(l_1 - 1)} \right) \otimes \cdots \otimes \left(\mathrm{I}^{(L_D)} - \mathrm{I}^{(L_D - 1)} \right) f$$

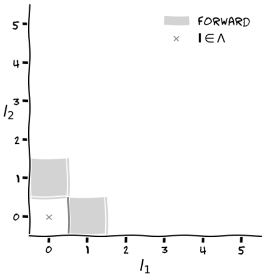
relies on "backward neighbours".

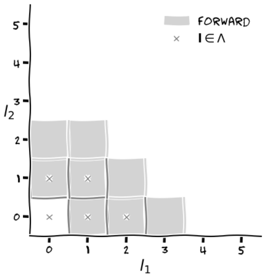
SC sparse grids: neighbours

- Forward neighbours of $\mathbf{l} \in \Lambda$:

$$\{\mathbf{l} + \mathbf{e}_i \mid 1 \leq i \leq D\}$$

$$\mathbf{e}_i := (0, \cdots, 0, 1, 0, \cdots, 0)$$



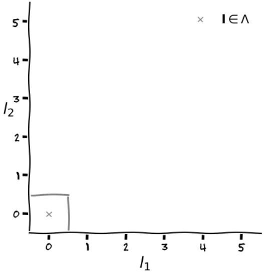


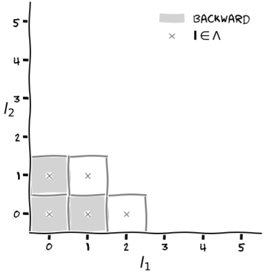
SC sparse grids: neighbours

- Backward neighbours of $\mathbf{l} \in \Lambda$:

$$\{\mathbf{l} - \mathbf{e}_i \mid l_i > 0,\ 1 \leq i \leq D\}$$

$$\mathbf{e}_i := (0, \cdots, 0, 1, 0, \cdots, 0)$$

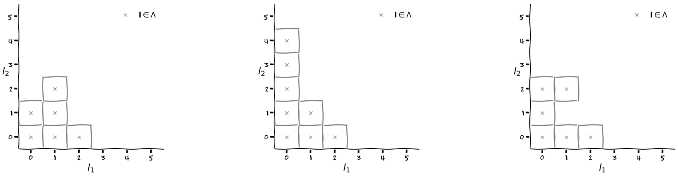




SC sparse grids: admissibility

- Admissible index set Λ:
  $\rightarrow$ all backward neighbours of Λ are in Λ

<u>Which of the following Λ are admissible?</u>

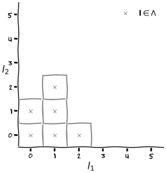


(a) (b) (c)

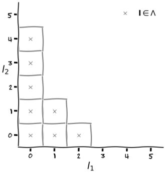
SC sparse grids: admissibility

- Admissible index set Λ:
  $\rightarrow$ all backward neighbours of Λ are in Λ

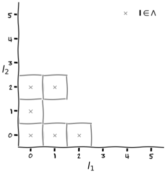
Which of the following Λ are admissible?



(a) No



(b) Yes

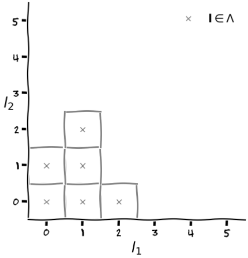


(c) No

SC sparse grids: admissibility

- Why "admissible"?

  $\rightarrow$ cannot construct all $\Delta^{(l_i)}$ if Λ is not admissible.



$$\Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f := \left(\mathrm{I}^{(l_1)} - \mathrm{I}^{(l_1-1)}\right) \otimes \cdots \otimes \left(\mathrm{I}^{(L_D)} - \mathrm{I}^{(L_D-1)}\right) f$$

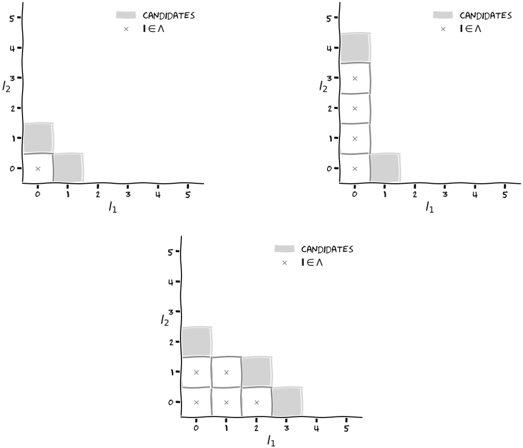
SC sparse grids: candidate directions

- Finally, given some D-dimensional $\mathrm{I}^{(\Lambda)} f(\mathbf{x})$
  $\rightarrow$ what directions of $\mathbf{x}$ are "candidates" for refinement?

SC sparse grids: candidate directions

- Finally, given some D-dimensional $\mathrm{I}^{(\Lambda)} f(\mathbf{x})$
  $\rightarrow$ what directions of $\mathbf{x}$ are "candidates" for refinement?

<u>Admissible forward neighbours that are not already in Λ</u>



SC sparse grids: candidate directions

- Recap, we have:

$$\mathrm{I}^{(\Lambda)} f = \sum_{\mathbf{l} \in \Lambda} \Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f(\mathbf{x})$$

  - Λ is admissible,
  - we can sample $\mathrm{I}^{(\Lambda)} f$'s admissible forward neighbours.

SC sparse grids: refinement in ND

- However:

$$\Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f := \left(\mathrm{I}^{(l_1)} - \mathrm{I}^{(l_1-1)}\right) \otimes \cdots \otimes \left(\mathrm{I}^{(L_D)} - \mathrm{I}^{(L_D-1)}\right) f$$

- contains many $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f$ terms for high D,
- yet many $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f$ terms in $\mathrm{I}^{(\Lambda)} f$ will cancel.

SC sparse grids: refinement in ND

- However:

$$\Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f := \left(\mathrm{I}^{(l_1)} - \mathrm{I}^{(l_1-1)}\right) \otimes \cdots \otimes \left(\mathrm{I}^{(L_D)} - \mathrm{I}^{(L_D-1)}\right) f$$

- contains many $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f$ terms for high D,
- yet many $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f$ terms in $\mathrm{I}^{(\Lambda)} f$ will cancel.

We can simply $\mathrm{I}^{(\Lambda)} f$ expression

SC sparse grids: combination coefficient

- Combination coefficient $c_{\mathbf{l}}$:

  $\rightarrow$ compute *a-priori* which $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f$ will survive cancellation ($c_{\mathbf{l}} \neq 0$).

$$
\begin{aligned}
\mathrm{I}^{(\Lambda)} f &= \sum_{\mathbf{l} \in \Lambda} \Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f(\mathbf{x}) \\
&= \boxed{\sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \, \mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f(\mathbf{x})}
\end{aligned}
$$

SC sparse grids: combination coefficient

- Combination coefficient $c_{\mathbf{l}}$:

  $\rightarrow$ compute *a-priori* which $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f$ will survive cancellation ($c_{\mathbf{l}} \neq 0$).

$$\mathrm{I}^{(\Lambda)} f = \sum_{\mathbf{l} \in \Lambda} \Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f(\mathbf{x})$$
$$= \boxed{\sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \, \mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f(\mathbf{x})}$$

- Advantage: <u>simple formulation</u>

  $\mathrm{I}^{(\Lambda)} f$ is just a linear combination of "standard" SC expansions $\mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f = \sum_{j_1} \cdots \sum_{j_D} f(x_{j_1}, \cdots, x_{j_D}) a_{j_1}^{(l_1)} \otimes \cdots \otimes a_{j_D}^{(l_D)}$.

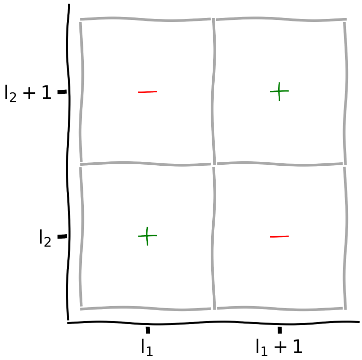
SC sparse grids: combination coefficient

- Combination coefficient:

$$\boxed{c_{\mathbf{l}} = \sum_{z_1=0}^{1} \cdots \sum_{z_d=0}^{1} (-1)^{\|\mathbf{z}\|_1} \chi(\mathbf{l} + \mathbf{z})}, \quad \text{where} \quad \chi(\mathbf{l}) = \begin{cases} 1 & \mathbf{l} \in \Lambda \\ 0 & \text{otherwise} \end{cases},$$

- in 2D: $\mathbf{z} \in \{(0,0), (1,0), (0,1), (1,1)\}$.
  $\rightarrow \mathbf{l} + \mathbf{z}$ the multi indices that affect $\mathbf{l}$'s contribution to $\mathrm{I}^{(\Lambda)} f$.

SC sparse grids: combination coefficient "game of life"



$\mathbf{l} + \mathbf{z}$	$\Delta^{(l_i)} \otimes \Delta^{(l_j)} f$	in/exclude $I^{(l_1)} \otimes I^{(l_2)} f$	$\|\mathbf{z}\|_1$
$(l_1 + 0, l_2 + 0)$	$\left(I^{(l_1)} - I^{(l_1-1)}\right) \otimes \left(I^{(l_2)} - I^{(l_2-1)}\right) f$	include	0
$(l_1 + 1, l_2 + 0)$	$\left(I^{(l_1+1)} - I^{(l_1)}\right) \otimes \left(I^{(l_2)} - I^{(l_2-1)}\right) f$	exclude	1
$(l_1 + 0, l_2 + 1)$	$\left(I^{(l_1)} - I^{(l_1-1)}\right) \otimes \left(I^{(l_2+1)} - I^{(l_2)}\right) f$	exclude	1
$(l_1 + 1, l_2 + 1)$	$\left(I^{(l_1+1)} - I^{(l_1)}\right) \otimes \left(I^{(l_2+1)} - I^{(l_2)}\right) f$	include	2

SC sparse grids: combination coefficient

- $(-1)^{\|\mathbf{z}\|_1}$ gives correct include/exclude (1/-1) for $\mathbf{l} + \mathbf{z}$

- Add $(-1)^{\|\mathbf{z}\|_1}$ to $c_{\mathbf{l}}$ $\underline{\text{if } \mathbf{l} + \mathbf{z} \in \Lambda}$.

[3] For efficient implementation see: W. Edeling, adaptive sparse-grid tutorial.

SC sparse grids: combination coefficient

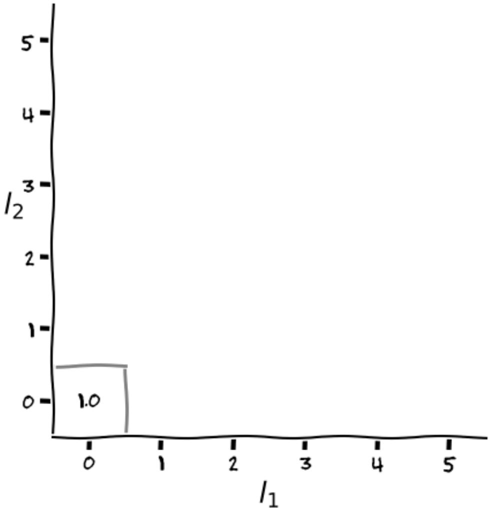
- $(-1)^{\|\mathbf{z}\|_1}$ gives correct include/exclude (1/-1) for $\mathbf{l} + \mathbf{z}$
- Add $(-1)^{\|\mathbf{z}\|_1}$ to $c_{\mathbf{l}}$ $\underline{\text{if } \mathbf{l} + \mathbf{z} \in \Lambda}$.
- Therefore [3]:

$$c_{\mathbf{l}} = \sum_{z_1=0}^{1} \cdots \sum_{z_d=0}^{1} (-1)^{\|\mathbf{z}\|_1} \, \chi\left(\mathbf{l} + \mathbf{z}\right), \quad \text{where} \quad \chi\left(\mathbf{l}\right) = \begin{cases} 1 & \mathbf{l} \in \Lambda \\ 0 & \text{otherwise} \end{cases},$$

[3] For efficient implementation see: W. Edeling, adaptive sparse-grid tutorial.

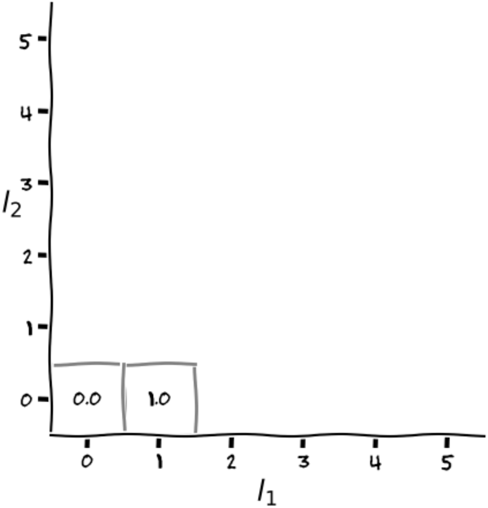
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



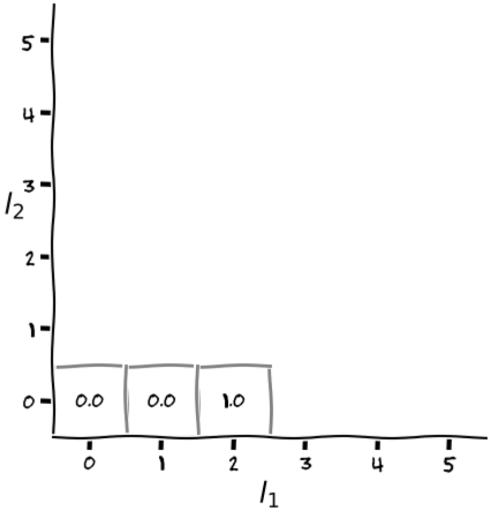
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



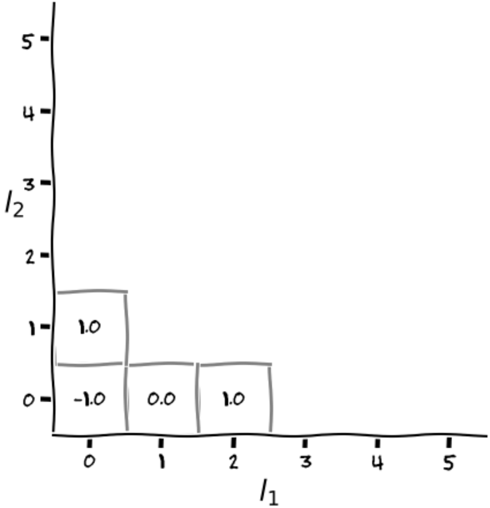
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



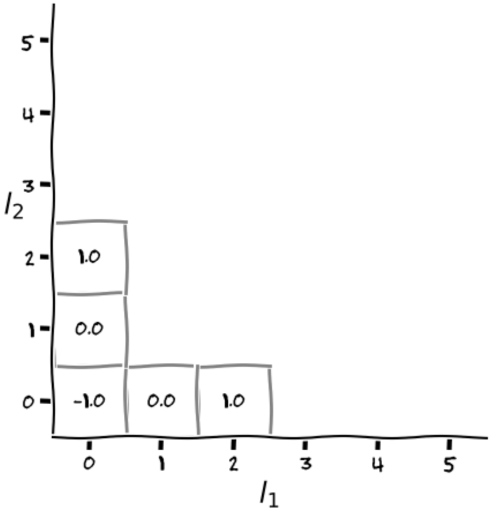
SC sparse grids: combination coefficient “game of life”

Multi indices with q:



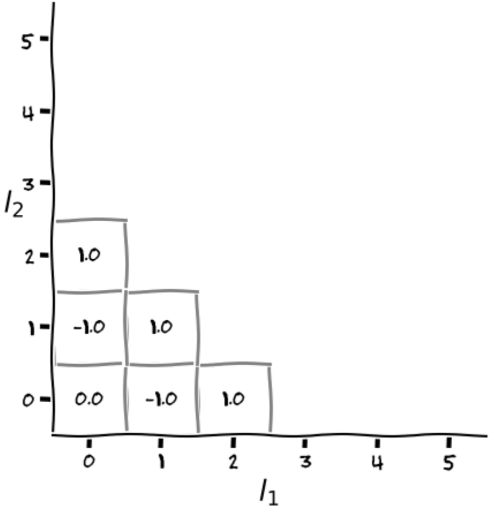
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



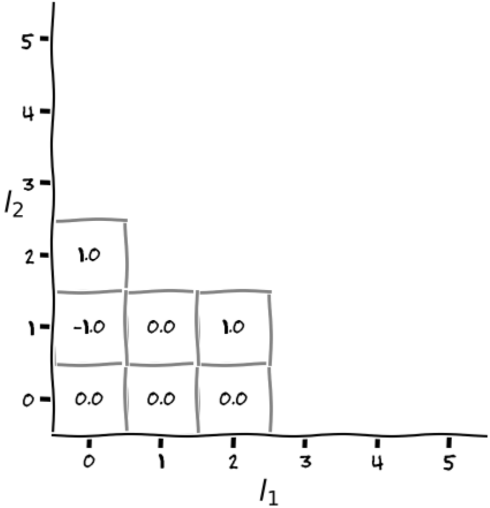
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



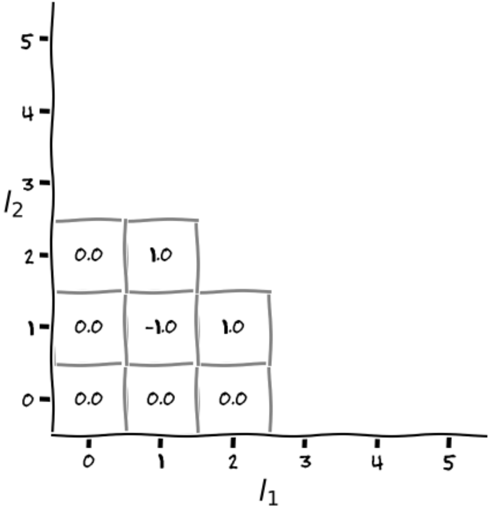
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



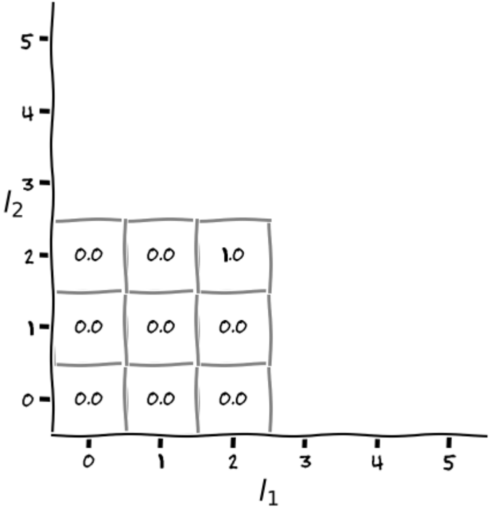
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



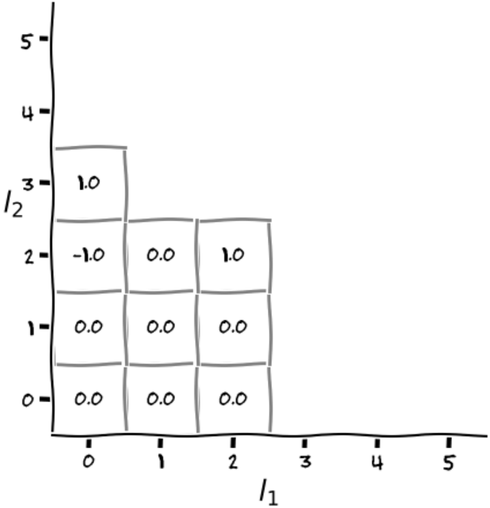
SC sparse grids: combination coefficient "game of life"

Multi indices with q:



SC sparse grids: combination coefficient “game of life”

Multi indices with q:



Isotropic SC sparse grids

Which important $\mathbf{i}$ do I admit into Λ?

[4] Also known as Smolyak sparse grids

Isotropic SC sparse grids

Which important $\mathbf{i}$ do I admit into Λ?

- Common choice: sparse-grid stochastic collocation
  $\rightarrow$ typically means isotropic grids[4], choose Λ as:

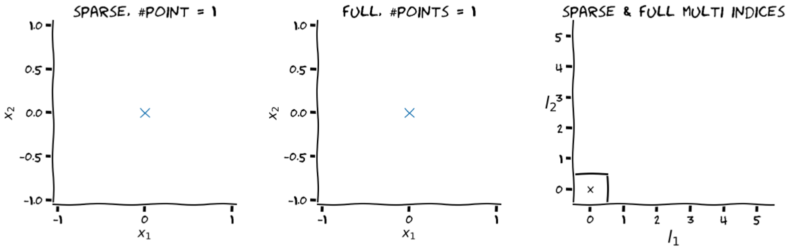
$$\Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq N\}$$

i.e. a standard "simplex set".

[4] Also known as Smolyak sparse grids

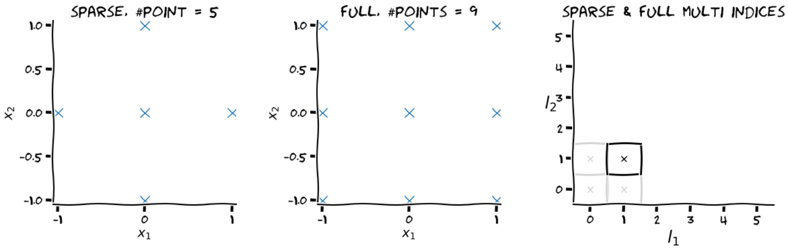
Isotropic SC sparse grids

$$\text{sparse } \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 0\} \;\; vs \;\; \text{full } \Lambda = \{(0,0)\}$$



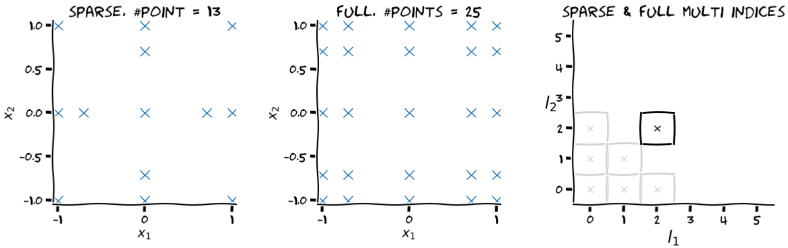
Isotropic SC sparse grids

$$\text{sparse} \;\; \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 1\} \;\; \textit{vs} \;\; \text{full} \;\; \Lambda = \{(1,1)\}$$



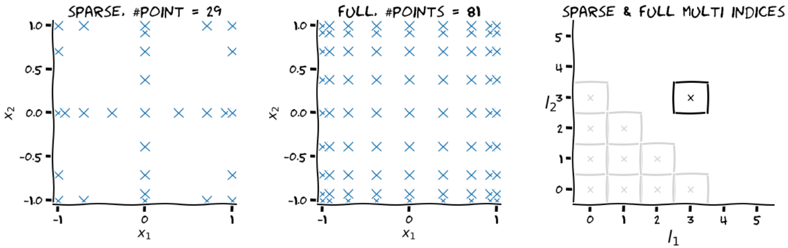
Isotropic SC sparse grids

$$\text{sparse } \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 2\} \;\; vs \;\; \text{full } \Lambda = \{(2,2)\}$$



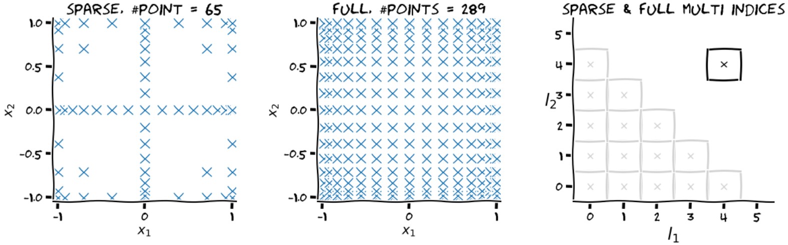
Isotropic SC sparse grids

$$\text{sparse } \ \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 3\} \ \ \textit{vs} \ \ \text{full } \ \Lambda = \{(3,3)\}$$



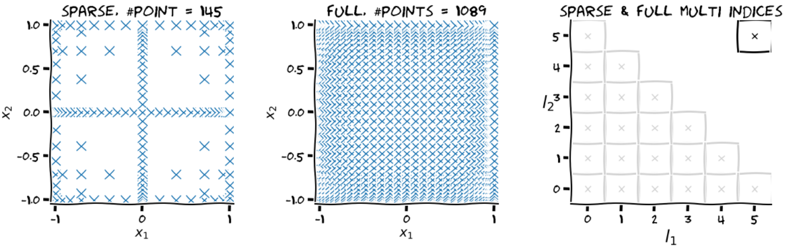
Isotropic SC sparse grids

$$\text{sparse} \;\; \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 4\} \;\; \mathit{vs} \;\; \text{full} \;\; \Lambda = \{4, 4)\}$$



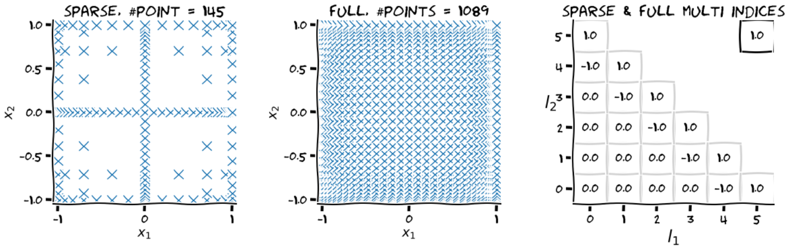
Isotropic SC sparse grids

$$\text{sparse} \;\; \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 5\} \;\; \textit{vs} \;\; \text{full} \;\; \Lambda = \{(5,5)\}$$



Isotropic SC sparse grids

$$\text{sparse}\;\; \Lambda = \{\mathbf{l} \mid |\mathbf{l}| \leq 5\} \;\; \textit{vs} \;\; \text{full}\;\; \Lambda = \{(5,5)\}$$

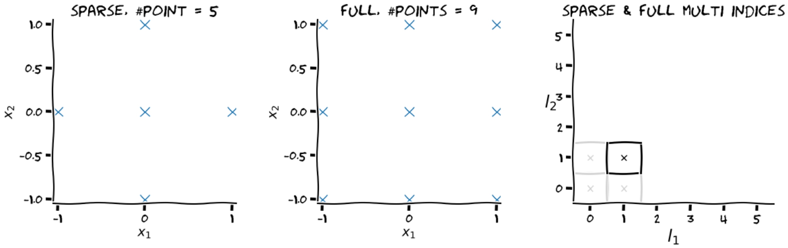


A linear combination $\sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}}\, \mathrm{I}^{(l_1)} \otimes \cdots \otimes \mathrm{I}^{(l_D)} f(\mathbf{x})$ of many low-order $\mathbf{l}$ yields less points than a single high-order $\mathbf{l}$.

Isotropic SC sparse grids

Which important **l** do **l** admit into Λ?

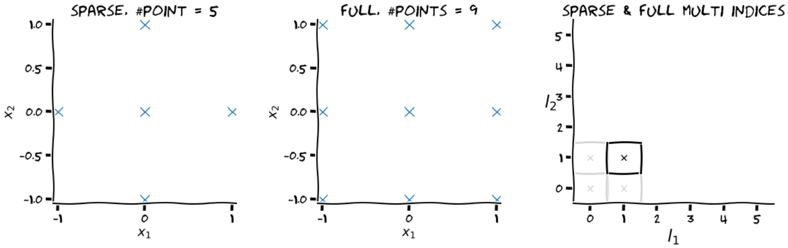
- What 'unimportant' effects are removed by this isotropic sparse grid:



Isotropic SC sparse grids

Which important **l** do **l** admit into Λ?

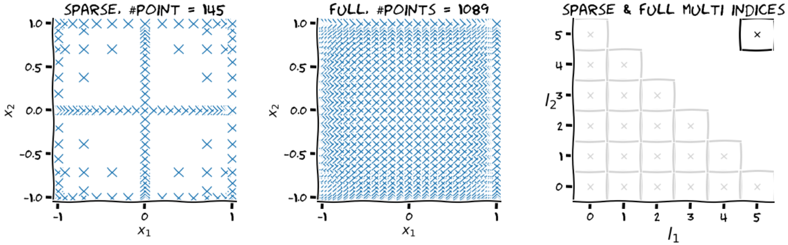
- What 'unimportant' effects are removed by this isotropic sparse grid:



- Full: polynomial representation up to $x_1^2 x_2^2$
- Sparse: only 1st order representation: x_1^2 or x_2^2

Isotropic SC sparse grids

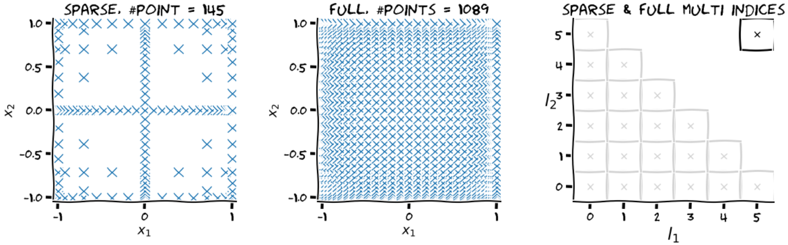
- Isotropic sparse grids assume (some) higher-order interaction effects are not important.



- Often ok in practice.

Isotropic SC sparse grids

- Isotropic sparse grids assume (some) higher-order interaction effects are not important.



- Often ok in practice.
- What assumption is made on relative importance of x_1 vs x_2?

Isotropic SC sparse grids: shortcomings

- Isotropic sparse grids assume all inputs are equally important.
- Often not true in practice:

  - small subset of $\{x_1, \cdots, x_D\}$ dominates output.
  - $f(\mathbf{x})$ has a small effective dimension $d < D$.

Isotropic SC sparse grids: shortcomings

- Isotropic sparse grids assume all inputs are equally important.
- Often <u>not</u> true in practice:

  - small subset of $\{x_1, \cdots, x_D\}$ dominates output.
  - $f(\mathbf{x})$ has a small effective dimension $d < D$.

- Dimension-adaptive sparse grids:
  - Hope effective dimension exists (really hope $d \ll D$).
  - <u>Challenge</u>: find effective dimension by iterative refinement.
  - Create anisotropic sampling plan.

Isotropic SC sparse grids: shortcomings

- Isotropic sparse grids assume all inputs are equally important.
- Often <u>not</u> true in practice:

  - small subset of $\{x_1, \cdots, x_D\}$ dominates output.
  - $f(\mathbf{x})$ has a small effective dimension $d < D$.

Isotropic SC sparse grids: shortcomings

- Isotropic sparse grids assume all inputs are equally important.
- Often not true in practice:

  - small subset of $\{x_1, \cdots, x_D\}$ dominates output.
  - $f(\mathbf{x})$ has a small effective dimension $d < D$.

- Dimension-adaptive sparse grids:
  - Hope effective dimension exists (really hope $d \ll D$).
  - Challenge: find effective dimension by iterative refinement.
  - Create anisotropic sampling plan.

Dimension-adaptive SC

- We already know how to do this

Dimension-adaptive SC

- We already know how to do this
- Iterative refinement:

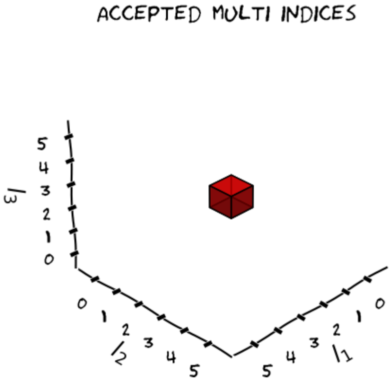
$$\mathrm{I}^{(\Lambda)}f = \sum_{\mathbf{l}\in\Lambda} \Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_D)} f(\mathbf{x})$$

$$= \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{j_1}^{N_{l_1}} \cdots \sum_{j_1}^{N_{l_D}} f\left(x_{j_1}^{(l_1)}, \cdots, x_{j_D}^{(l_D)}\right) a_{j_1}^{(l_1)}(x_1) \otimes \cdots \otimes a_{j_D}^{(l_D)}(x_D)$$

- Build Λ on-the-fly.

Dimension-adaptive SC: algorithm

- Start: $\Lambda = \{0, 0, \cdots, 0\}$
- Look ahead: compute candidate $\mathbf{l}$ (sampling $f(\mathbf{x})$)
- Adapt: accept 1 candidate $\mathbf{l}$ into Λ
- Repeat.



Dimension-adaptive SC: algorithm

- How to select 1 candidate $\mathbf{l}$?
  $\rightarrow$ define error metric to identify "important" $\mathbf{l}$.
  $\rightarrow$ Common choice: hierarchical surplus:

$$s\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right) := f\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right) - I^{(\Lambda)} f\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right), \quad \mathbf{x}_{\mathbf{j}}^{(\mathbf{l})} \in X_{\mathbf{l}} \backslash X_{\Lambda}.$$

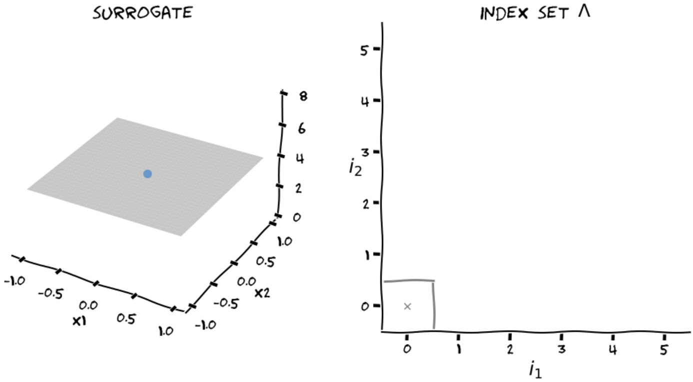
- Mean surplus error for a candidate $\mathbf{l}$:

$$e^{(\mathbf{l})} := \frac{1}{\#(X_{\mathbf{l}} \backslash X_{\Lambda})} \sum_{\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})} \in X_{\mathbf{l}} \backslash X_{\Lambda}} \left\| s\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right) \right\|_2.$$

- $\Lambda \leftarrow \Lambda \cup \mathbf{l}^*$ where $\mathbf{l}^* = \{\mathbf{l} \mid \max(e^{(\mathbf{l})})\}$

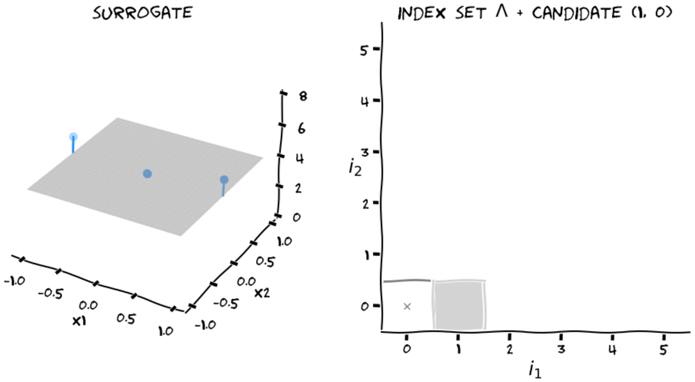
Dimension-adaptive SC: algorithm

- Graphically:



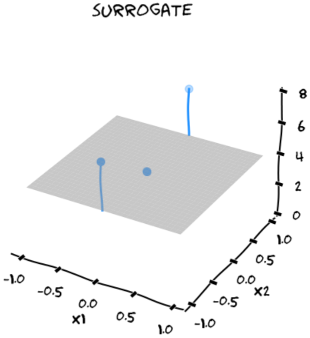
Dimension-adaptive SC: algorithm

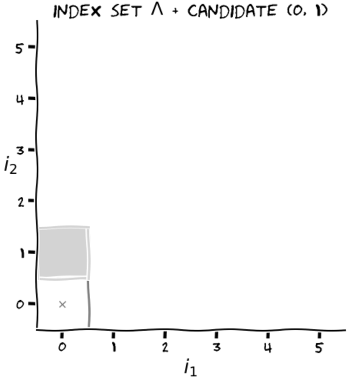
- Graphically:



Dimension-adaptive SC: algorithm

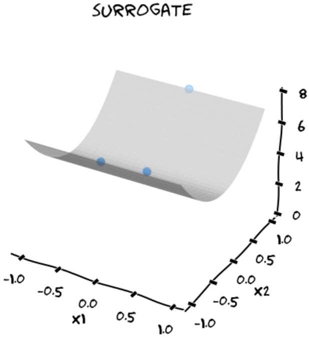
- Graphically:

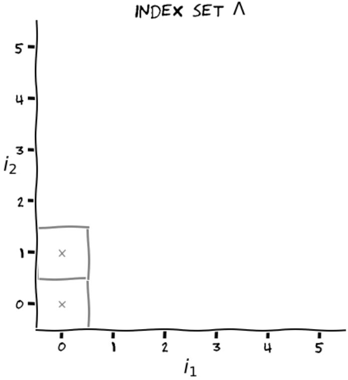




Dimension-adaptive SC: algorithm

- Graphically:





Dimension-adaptive SC: algorithm

- Hierarchical surplus is not only option.
- Alternative error: choose candidate $\mathbf{l}$ with largest jump in $\mathbb{V}\mathrm{ar}[f(\mathbf{x})]$.

Dimension-adaptive SC: algorithm

- Hierarchical surplus is not only option.
- Alternative error: choose candidate $\mathbf{l}$ with largest jump in $\mathbb{V}\mathrm{ar}[f(\mathbf{x})]$.
- What is a downside of (all) refinement errors?

Dimension-adaptive SC: algorithm

- Hierarchical surplus is not only option.
- Alternative error: choose candidate $\mathbf{l}$ with largest jump in $\mathbb{V}\mathrm{ar}[f(\mathbf{x})]$.
- What is a downside of (all) refinement errors?
- You are creating sampling plan for a specific output $f(x)$.

Dimension-adaptive SC: Recap

- Hope effective dimension $\ll D$.
- Be sure about your output QoI.
- Begin iterative sampling:

  - Start: $\Lambda = \{0, 0, \cdots, 0\}$
  - Look ahead: compute candidate $\mathbf{l}$ (sampling $f(\mathbf{x})$)
  - Adapt: accept 1 candidate $\mathbf{l}$ into Λ based on refinement error (surplus)
  - Repeat.

- Stop (e.g. when budget runs out).
- Post process results: mean, var and sensitivity analysis.

Sobol indices

Sensitivity analysis

- Sobol sensitivity indices:
  - which (subset of) parameters are most important?
  - obtained by post-processing ensemble.
  - global, variance-based sensitivity indices.
  - based on High Dimensional Model Representation (HDMR) decomposition.

Sobol representation (HDMR)

Assume for now all $x_i \sim \mathcal{U}[0,1]$, domain $\mathbf{x} \in \Gamma = [0,1]^D$

HDMR expansion:

$$f(\mathbf{x}) = f_0 + \sum_i f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \sum_{i<j<k} f_{ijk}(x_i, x_j, x_k) + ...$$

$$... + f_{1,2,..,D}(x_1, ..., x_D)$$

Note that:

- f_i: 1st-order effect, contribution of single inputs.
- f_{ij}: 2nd-order interaction effect, varying 2 inputs simultaneously.

Sobol representation (HDMR)

Assume for now all $x_i \sim \mathcal{U}[0,1]$, domain $\mathbf{x} \in \Gamma = [0,1]^D$

HDMR expansion:

$$f(\mathbf{x}) = f_0 + \sum_i f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) + \sum_{i<j<k} f_{ijk}(x_i, x_j, x_k) + ...$$

$$... + f_{1,2,..,D}(x_1, ..., x_D)$$

Note that:

- f_i: 1st-order effect, contribution of single inputs.
- f_{ij}: 2nd-order interaction effect, varying 2 inputs simultaneously.

Sobol representation: 2nd order

Second order approximation:

$$f(\mathbf{x}) \approx f_0 + \sum_{i=1}^{D} f_i(X_i) + \sum_{j=2}^{D} \sum_{i=1}^{j-1} f_{ij}(X_i, X_j)$$

e.g. if $D = 3$, $f \approx f_0 + f_1 + f_2 + f_3 + f_{12} + f_{13} + f_{23}$.

assuming higher-order interaction terms have negligible impact.

Need constraints (on $f_0, f_i, f_{ij}, \ldots$) to make this representation unique.

Sobol representation: constraints

Constraints:

$$\int_0^1 f_i(x_i)\,dx_i = 0,\ \int_0^1 f_{ij}(x_i, x_j)\,dx_i = 0, \int_0^1 f_{ij}(x_i, x_j)\,dx_j = 0 \ \ \forall\, i, j$$

Terms $f_0, f_i, f_{ij}, ...$ can be obtained by minimizing

$$\int_\Gamma [f(\mathbf{x}) - \tilde{f}(\mathbf{x})]^2 d\mathbf{x}$$

under these constraints

(where $\tilde{f}(\mathbf{x}) = f_0 + \sum_{i=1}^{D} f_i\left(x_i\right) + \sum_{j=2}^{D}\sum_{i=1}^{j-1} f_{ij}\left(x_i, x_j\right)$)

Sobol representation: expansion terms

Resulting terms:

$$f_0 = \int_{\Gamma} f(\mathbf{x})\, d\mathbf{x}$$

$$f_i(x_i) = \int_{\Gamma^{D-1}} f(\mathbf{x})\, dx_{\sim i} - f_0$$

$$f_{ij}(x_i, x_j) = \int_{\Gamma^{D-2}} f(\mathbf{x})\, dx_{\sim ij} - f_i(x_i) - f_j(x_j) - f_0$$

notation:
$\Gamma = [0,1]^D, \Gamma^{D-1} = [0,1]^{D-1}, \Gamma^{D-2} = [0,1]^{D-2}$,
$dx_{\sim i} = dx_1 ... dx_{i-1} dx_{i+1} ... dx_D$, similar for $dx_{\sim ij}$

Sobol representation: orthogonality

Result of constraints: terms are **orthogonal**: for all i, j, k,

$$\int_\Gamma f_0\, f_i(x_i)\, d\mathbf{x} = 0$$
$$\int_\Gamma f_0\, f_{ij}(x_i, x_j)\, d\mathbf{x} = 0$$
$$\int_\Gamma f_k(x_k)\, f_{ij}(x_i, x_j)\, d\mathbf{x} = 0$$

Use the square of the component functions for sensitivity analysis (SA).

Variance-based SA

- Variance:

$$D := \mathbb{V}\mathrm{ar}[f] = \int_\Gamma (f - f_0)^2 d\mathbf{x}$$

- Insert HDMR expansion for f:

$$D = \int_\Gamma \left(\sum_i f_i + \sum_{i<j} f_{ij} \right)^2 d\mathbf{x}$$
$$= ?$$

Variance-based SA

- Variance:

$$D := \mathbb{V}\mathrm{ar}[f] = \int_\Gamma (f - f_0)^2 d\mathbf{x}$$

- Insert HDMR expansion for f:

$$\begin{aligned} D &= \int_\Gamma \left(\sum_i f_i + \sum_{i<j} f_{ij} \right)^2 d\mathbf{x} \\ &= \sum_i \int_\Gamma f_i^2 d\mathbf{x} + \sum_{i<j} \int_\Gamma f_{ij}^2 d\mathbf{x} \\ &= \sum_i \int f_i^2 dx_i + \sum_{i<j} \int f_{ij}^2 dx_i dx_j \end{aligned}$$

- D = sum of squared integrated component functions

Variance-based SA

- Squared integrated component functions = "partial variances":

$$D_i := \int f_i^2 dx_i, \qquad D_{ij} := \int f_{ij}^2 dx_i dx_j$$

- Such that:

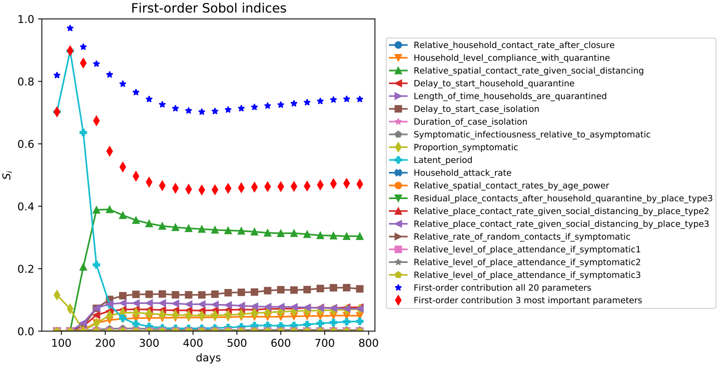
$$D := \sum_i D_i + \sum_{i<j} D_{ij} + (\text{higher order terms})$$

- Sobol indices: divide by total variance:

$$\boxed{1 = \sum_i S_i + \sum_{i<j} S_{ij} + (\text{higher order terms})}$$

Variance-based SA

- First order Sobol indices S_i:
  - simply percentage of variance due to x_i alone
  - paint intuitive picture of sensitivity, e.g.[5]:



[5] Edeling, Wouter, et al. "The impact of uncertainty on predictions of the CovidSim epidemiological code." Nature Computational Science 1.2 (2021)

Variance-based SA

- How to compute $S_i := D_i/D$?
  - Using Monte Carlo [6]
  - Using Stochastic Collocation [7]
  - Using Polynomial Chaos (most elegant) [8]

[6] Saltelli, Andrea, et al. Sensitivity analysis in practice: a guide to assessing scientific models. Vol. 1., 2004.

[7] Tang, Gary, et al. "Global sensitivity analysis for stochastic collocation expansion." CSRI Summer Proceedings 2009 (2010)

[8] Sudret, Bruno. "Global sensitivity analysis using polynomial chaos expansions." Reliability engineering & system safety 93.7 (2008)

Sobol indices with PCE

- Remember the multivariate PCE expansion:

$$f(\mathbf{x}) = \sum_{\mathbf{i}\in\Lambda} \hat{f}_{\mathbf{i}} \Phi_{\mathbf{i}}(\mathbf{x})$$

with:

$$\hat{f}_{\mathbf{i}} := \frac{\mathbb{E}\left[f \Phi_{\mathbf{i}}\right]}{\gamma_{\mathbf{i}}}, \quad \gamma_{\mathbf{i}} := \mathbb{E}\left[\Phi_{\mathbf{i}}\Phi_{\mathbf{i}}\right]$$

- We got the variance "for free":

$$\mathbb{E}[f] = \hat{f}_{\mathbf{0}}, \quad D := \mathbb{V}\mathrm{ar}[f] = \sum_{\substack{\mathbf{i}\in\Lambda \\ \mathbf{i}\neq\mathbf{0}}} \hat{f}_{\mathbf{i}}^2 \gamma_{\mathbf{i}}$$

Sobol indices with PCE

- Partial variances are also easily found from PCE coefficients:

$$D_{\mathbf{u}} = \sum_{\mathbf{k}\in\mathcal{K}_{\mathbf{u}}} \hat{f}_{\mathbf{k}}^2 \gamma_{\mathbf{k}} \ \text{with}$$

$$\mathcal{K}_{\mathbf{u}} = \{\mathbf{k} \mid k_i > 0 \ \text{if} \ \ k_i \in \mathbf{u}, \ \ k_j = 0 \ \text{otherwise}\}$$

$\mathcal{K}_{\mathbf{u}}$ = set of all multi indices that vary *only* the inputs indexed by $\mathbf{u}$

Sobol indices with PCE

- $\mathcal{K}_\mathbf{u} = \{\mathbf{k} \mid k_i > 0 \text{ if } k_i \in \mathbf{u}, \quad k_j = 0 \text{ otherwise}\}$
  Which indices are in $\mathcal{K}_\mathbf{u}$ if $u = \{1\}$ (Sobol index S_1)?

$\lvert\mathbf{i}\rvert$	Multi-index $\mathbf{i}$
0	(0 0 0 0)
1	(1 0 0 0)
	(0 1 0 0)
	(0 0 1 0)
	(0 0 0 1)
2	(2 0 0 0)
	(1 1 0 0)
	(1 0 1 0)
	(1 0 0 1)
	(0 2 0 0)
	(0 1 1 0)
	(0 1 0 1)
	(0 0 2 0)
	(0 0 1 1)
	(0 0 0 2)

Sobol indices with PCE

- $\mathcal{K}_{\mathbf{u}} = \{\mathbf{k} \mid k_i > 0 \text{ if } k_i \in \mathbf{u}, \quad k_j = 0 \text{ otherwise}\}$
  Which indices are in $\mathcal{K}_{\mathbf{u}}$ if $u = \{1\}$ (Sobol index S_1)?

$\lvert\mathbf{i}\rvert$	Multi-index $\mathbf{i}$
0	(0 0 0 0)
1	(1 0 0 0)
	(0 1 0 0)
	(0 0 1 0)
	(0 0 0 1)
2	(2 0 0 0)
	(1 1 0 0)
	(1 0 1 0)
	(1 0 0 1)
	(0 2 0 0)
	(0 1 1 0)
	(0 1 0 1)
	(0 0 2 0)
	(0 0 1 1)
	(0 0 0 2)

Sobol indices with PCE

- $\mathcal{K}_{\mathbf{u}} = \{\mathbf{k} \mid k_i > 0 \ \text{if} \ \ k_i \in \mathbf{u}, \ \ k_j = 0 \ \ \text{otherwise}\}$
  Which indices are in $\mathcal{K}_{\mathbf{u}}$ if $u = \{1, 3\}$ (Sobol index S_{13})?

$\lvert\mathbf{i}\rvert$	Multi-index $\mathbf{i}$
0	(0 0 0 0)
1	(1 0 0 0)
	(0 1 0 0)
	(0 0 1 0)
	(0 0 0 1)
2	(2 0 0 0)
	(1 1 0 0)
	(1 0 1 0)
	(1 0 0 1)
	(0 2 0 0)
	(0 1 1 0)
	(0 1 0 1)
	(0 0 2 0)
	(0 0 1 1)
	(0 0 0 2)

Sobol indices with sparse-grid SC

- What if we have an (anisotropic) sparse-grid SC expansion?

$$\mathrm{I}^{(\Lambda)} f = \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}) \mathbf{a}_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x})$$

- Each standard SC expansion can be presented <u>exactly</u> by a PCE (both are polynomials):

$$\sum_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}) \mathbf{a}_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x}) = \sum_{\mathbf{k} \in \Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \Phi_{\mathbf{k}}(\mathbf{x})$$

Sobol indices with sparse-grid SC

- What if we have an (anisotropic) sparse-grid SC expansion?

$$\mathrm{I}^{(\Lambda)}f = \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})})\mathbf{a}_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x})$$

- Each standard SC expansion can be presented exactly by a PCE (both are polynomials):

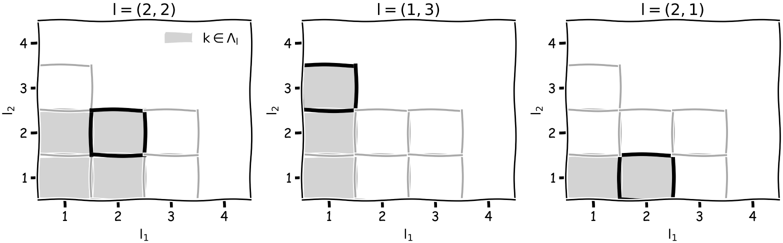
$$\sum_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})})\mathbf{a}_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x}) = \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})}\Phi_{\mathbf{k}}(\mathbf{x})$$

- Select all indices up to $\mathbf{l}$ to do so:

$$\Lambda_{\mathbf{l}} := \{\mathbf{k} \mid \mathbf{k} \leq \mathbf{l},\ \mathbf{l} \in \Lambda\}.$$

Sobol indices with sparse-grid SC

- Examples of $\Lambda_{\mathbf{l}} = \{\mathbf{k} \mid \mathbf{k} \leq \mathbf{l},\ \mathbf{l} \in \Lambda\}$:



Sobol indices with sparse-grid SC

- Then we have:

$$
\begin{aligned}
\mathrm{I}^{(\Lambda)} f &= \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{\mathbf{j}} f(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}) \mathbf{a}_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x}) \\
&= \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{\mathbf{k} \in \Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \Phi_{\mathbf{k}}(\mathbf{x})
\end{aligned}
$$

- Let's move $c_{\mathbf{l}}$ into second summation
  $\rightarrow$ expand $+$ group by like $\mathbf{k}$

Sobol indices with sparse-grid SC

$$\mathrm{I}^{(\Lambda)} f = \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \Phi_{\mathbf{k}}(\mathbf{x})$$

- Expand + group by like **k**: $\Lambda = \{(0,0),(1,0),(0,1),(1,1)\}$

$$I^{(\Lambda)} f = c_{00}\left(\hat{f}_{00}^{00}\Phi_{00}\right) + c_{01}\left(\hat{f}_{00}^{01}\Phi_{00} + \hat{f}_{01}^{01}\Phi_{01}\right) + c_{10}\left(\hat{f}_{00}^{10}\Phi_{00} + \hat{f}_{10}^{10}\Phi_{10}\right) + c_{11}\left(\hat{f}_{00}^{11}\Phi_{00} + \hat{f}_{01}^{11}\Phi_{01} + \hat{f}_{10}^{11}\Phi_{10} + \hat{f}_{11}^{11}\Phi_{11}\right)$$

$$= \left(c_{00}\hat{f}_{00}^{00} + c_{01}\hat{f}_{00}^{01} + c_{10}\hat{f}_{00}^{10} + c_{11}\hat{f}_{00}^{11}\right)\Phi_{00} + \left(c_{01}\hat{f}_{01}^{01} + c_{11}\hat{f}_{01}^{11}\right)\Phi_{01} + \left(c_{10}\hat{f}_{10}^{10} + c_{11}\hat{f}_{10}^{11}\right)\Phi_{10} + \left(c_{11}\hat{f}_{11}^{11}\right)\Phi_{11}$$

<u>$c_{\mathbf{l}}$ moves into 2nd summation.</u>

Sobol indices with sparse-grid SC

$$
\begin{aligned}
\mathrm{I}^{(\Lambda)} f &= \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \Phi_{\mathbf{k}}(\mathbf{x}) \\
&= \sum_{\mathbf{l}\in\Lambda} \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}^{-1}} c_{\mathbf{k}} \hat{f}_{\mathbf{l}}^{(\mathbf{k})} \Phi_{\mathbf{l}}(\mathbf{x})
\end{aligned}
$$

- Remember: $\Lambda_{\mathbf{l}} = \{\mathbf{k} \mid \mathbf{k} \le \mathbf{l},\ \mathbf{l} \in \Lambda\}$
  $\rightarrow \Lambda_{\mathbf{l}}^{-1}$ is:

$$\Lambda_{\mathbf{l}}^{-1} := \{\mathbf{k} \mid \mathbf{k} \ge \mathbf{l},\ \mathbf{l} \in \Lambda\}.$$

What is the consequence?

Sobol indices with sparse-grid SC

$$
\begin{aligned}
\mathrm{I}^{(\Lambda)} f &= \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \Phi_{\mathbf{k}}(\mathbf{x}) \\
&= \sum_{\mathbf{l}\in\Lambda} \underbrace{\sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}^{-1}} c_{\mathbf{k}} \hat{f}_{\mathbf{l}}^{(\mathbf{k})}}_{:=\hat{g}_{\mathbf{l}}} \Phi_{\mathbf{l}}(\mathbf{x}) \\
&= \sum_{\mathbf{l}\in\Lambda} \hat{g}_{\mathbf{l}} \Phi_{\mathbf{l}}(\mathbf{x})
\end{aligned}
$$

Sobol indices with sparse-grid SC

$$
\begin{aligned}
\mathrm{I}^{(\Lambda)} f &= \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \Phi_{\mathbf{k}}(\mathbf{x}) \\
&= \sum_{\mathbf{l}\in\Lambda} \underbrace{\sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}^{-1}} c_{\mathbf{k}} \hat{f}_{\mathbf{l}}^{(\mathbf{k})}}_{:=\hat{g}_{\mathbf{l}}} \Phi_{\mathbf{l}}(\mathbf{x}) \\
&= \sum_{\mathbf{l}\in\Lambda} \hat{g}_{\mathbf{l}} \Phi_{\mathbf{l}}(\mathbf{x})
\end{aligned}
$$

$I^{(\Lambda)} f$ can be written in standard PCE form, use standard PCE Sobol calculation

Conclusion: everything is a standard PCE

$$
\begin{aligned}
I^{(\Lambda)} f &= \sum_{\mathbf{l} \in \Lambda} \Delta^{(l_1)} \otimes \cdots \otimes \Delta^{(l_d)} f \\
&= \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{j_1=1}^{N_1} \cdots \sum_{j_d=1}^{N_1} f\left(\mathbf{x}_{\mathbf{j}}^{(\mathbf{l})}\right) a_{\mathbf{j}}^{(\mathbf{l})}(\mathbf{x}) \\
&= \sum_{\mathbf{l} \in \Lambda} c_{\mathbf{l}} \sum_{\mathbf{k} \in \Lambda_{\mathbf{l}}} \hat{f}_{\mathbf{k}}^{(\mathbf{l})} \phi_{\mathbf{k}}(\mathbf{x}) = \sum_{\mathbf{l} \in \Lambda} \sum_{\mathbf{k} \in \Lambda_{\mathbf{l}}^{-1}} c_{\mathbf{k}} \hat{f}_{\mathbf{l}}^{(\mathbf{k})} \phi_{\mathbf{l}}(\mathbf{x}) \\
&= \sum_{\mathbf{l} \in \Lambda} \hat{g}^{(\mathbf{l})} \phi_{\mathbf{l}}(\mathbf{x}),
\end{aligned}
$$

= PCE & (standard/sparse isotropic/dimension adaptive) SC