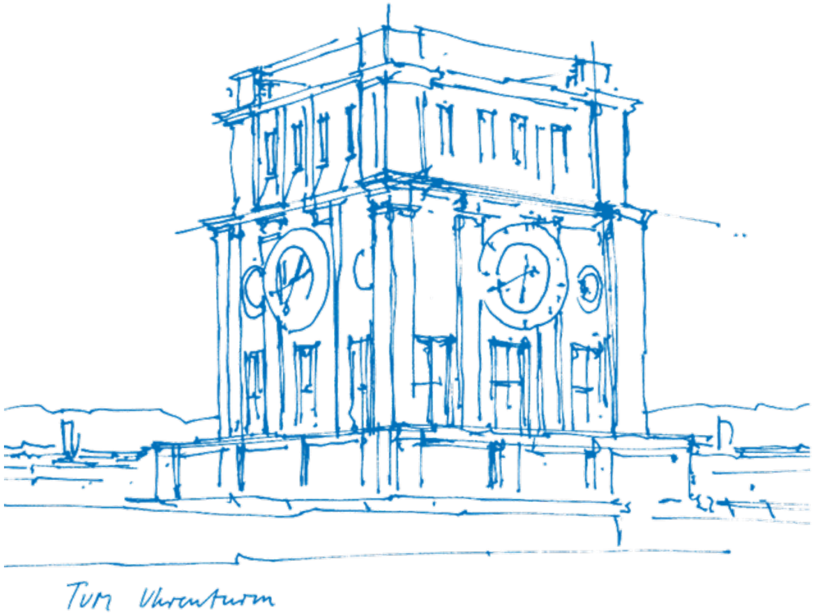
Efficient FUQ and SA with Spatially Adaptive SG

Ivana Jovanovic Buha

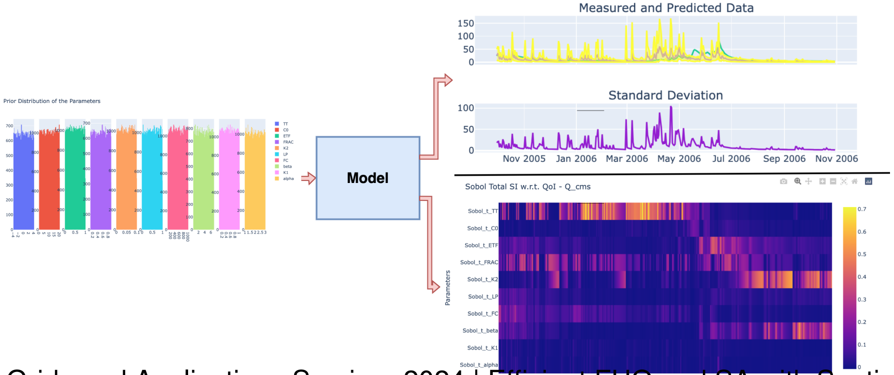
Technical University of Munich
TUM School of Computation, Information and Technology
Chair of Scientific Computing

Bonn, Germany, 11. September 2024



Motivation

- **Scientific Approach:** Experiment with different strategies for combining (Spatially Adaptive) Sparse Grids (SG) with Uncertainty Quantification (UQ) and Sensitivity Analysis (SA) algorithms
- **Final Goal:**
  Efficient UQ and SA of complex dynamical models (e.g., HBV-SASK hydrologic model [1]) by utilizing (adaptive) Sparse Grids
- **Impediments:**
  - High-dimensionality
  - High (model) execution time
  - Model as a black box
  - Possible discontinuities in the parameter space; anisotropic or decoupled parameters
  - Output of the model - time signal



UQ:

$$\mathbb{E}[\mathcal{O}(f)] := \int_{\Gamma} \mathcal{O}(f)(t, \boldsymbol{\theta})\rho(\boldsymbol{\theta})d\boldsymbol{\theta}$$

$$Var[\mathcal{O}(f)] := \mathbb{E}[\mathcal{O}(f)^2] - (\mathbb{E}[\mathcal{O}(f)])^2$$

Variance-based SA [Sobol 2001]:

$$S_j := \frac{Var[f] - \mathbb{E}_{\theta_j}[Var_{\boldsymbol{\theta}_{\sim j}}[f|\theta_j]]}{Var[f]}$$

Building Blocks

- **Sparse Grid (SG)**
  - Standard SG and Combination Technique
  - Spatially Adaptive SG
  - SparseSpACE Framework
- **Non-intrusive UQ and SA**
- **UQ with SG**
  - Different variant
  - Initial results
- **UQ with SG for Time-dependent models**
  - Initial results
  - Future works

Sparse Grid

Main problem: With high-dimensional problems the number of grid points in a regular grid increases exponentially

Sparse Grids Idea [4]: Selecting points that contribute most to the solution given certain smoothness criteria
[⇒] Reduction of point numbers from $\mathcal{O}(N^d)$ to $\mathcal{O}(Nlog(N)^{d-1})$

Sparse Grids can be constructed in various ways:

- hierarchization, combination technique, adaptivity...
- different basis functions (e.g., linear hat, Lagrange poly, b-splines etc.)
- or the point positions/ grid types (e.g., equidistant grids, Clenshaw-Curtis, Leja)

Sparse Grid & Combination Technique

Combination Technique (CT) [5]:

Efficient SG computation by linearly combining computations on cheap/coarser anisotropic full grids (e.g., component grids)

For these full grids any conventional full grid solver can be applied

$$u_{\mathscr{I}}^{CT} = \sum_{\boldsymbol{l}\in\mathscr{I}} c_{\boldsymbol{l}} u_{\boldsymbol{l}}^d = \sum_{\boldsymbol{l}\in\mathscr{I}} \sum_{\boldsymbol{l}\leq\boldsymbol{i}\leq\boldsymbol{l}+1, \boldsymbol{i}\in\mathscr{I}} (-1)^{\|\boldsymbol{i}-\boldsymbol{l}\|_1} u_{\boldsymbol{l}}^d, \quad \mathscr{I} = \{\boldsymbol{l}\in\mathbb{N}^d \mid \|\boldsymbol{l}\|_1 = l+d-1\} \tag{1}$$

$c_{\boldsymbol{l}}$ scalar coefficients streaming from the combinatorics of the difference formulation

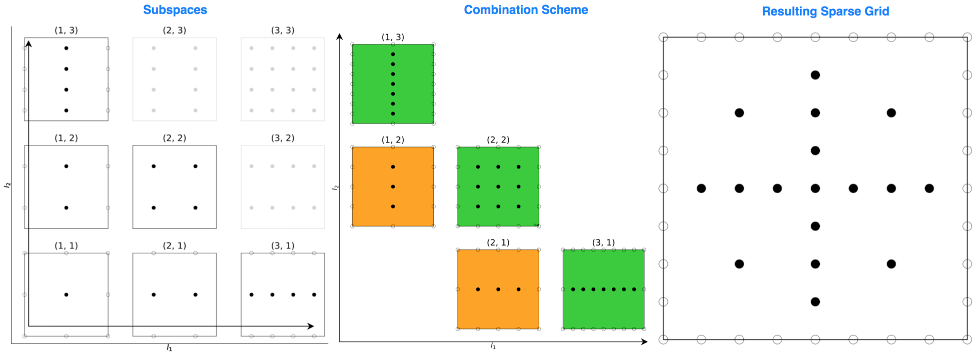


Abbildung: Combination technique represented via subspaces, grid components and the final resulting sparse grid; green component grids are added and orange ones are subtracted

Spatially Adaptive Sparse Grid

General assumptions of standard SG - similar contribution of all dimensions to the result and an overall smoothness throughout the domain

- **Dimension Adaptivity** [6, 7] - different dimensions or interactions between them contribute in different magnitudes to the solution; adjusting the index set in CT
- **Spatial Adaptivity** - often different resolutions are required at different parts of the domain; do not add full subspaces but only specific points to the SG
  - **Drawback:** Standard CT offers no spatial adaptivity
  - **CT with Spatial Adaptivity** - use rectilinear grids constructed via a tensor product of refined 1-D grids [2]

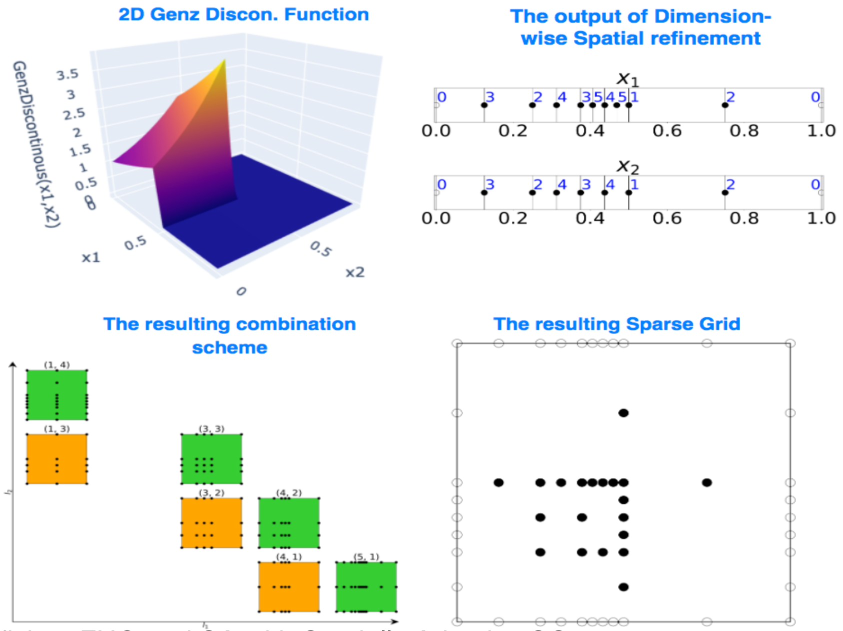
Spatially Adaptive Sparse Grid

Combination Technique with Dimension-Wise Refinement [2]

CT with Spatial Adaptivity - use rectilinear grids constructed via a tensor product of refined 1-D grids

Key components:

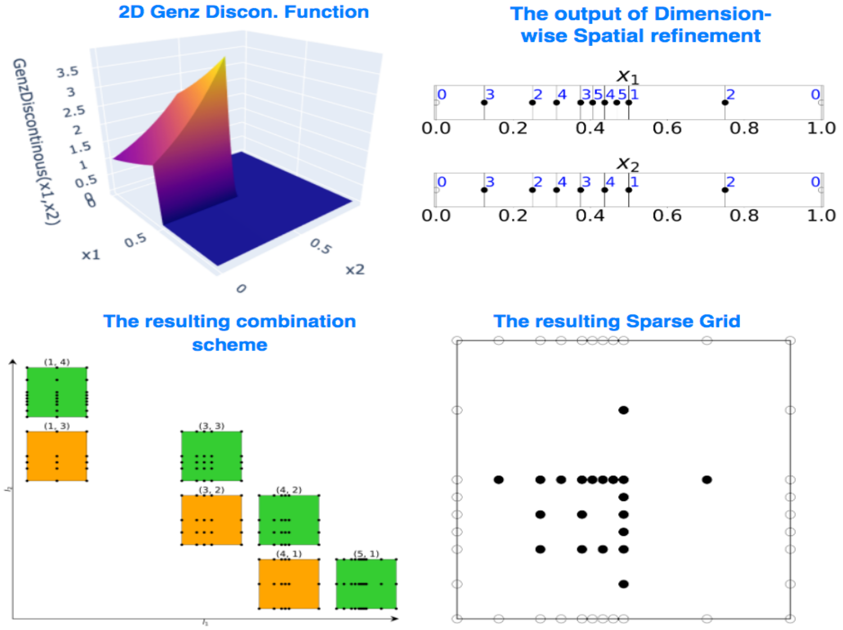
- 1D refinements define the adaptive process
- Creating a global valid combination scheme from 1D refinements
- Special error estimators guide the refinement
- Data structure and tree rebalancing for better performance
- SparseSpACE Framework
  https://github.com/obersteiner/sparseSpACE



Spatially Adaptive SG CT & SparseSpACE Framework

Key components:

- Step 1: 1D refinements define the adaptive process
  - look at every child in a grid and refine based on error approximation
  - error estimator
    - compute for each leaf node $p \in \boldsymbol{P}^k$, for each dim. $k \in [d]$ over all grids
    - for grid $\boldsymbol{l}$ error estimate $\varepsilon_p^{k,\boldsymbol{l}}$ is 1D surplus value weighted by the volume of the respective basis function
    - $\varepsilon_p^k = \sum_{\boldsymbol{l} \in I} |c_{\boldsymbol{l}} \cdot \varepsilon_p^{k,\boldsymbol{l}}|$
    - refine every point p with error estimate $|\varepsilon_p^k| \geq \gamma \cdot \varepsilon^{\max}$ (e.g., $\gamma = 0.5$)
    - global error - $\varepsilon = \sum_{k=1}^{d} \sum_{j=1}^{|\boldsymbol{P}^k|} |\varepsilon_j^k|$
  - output [$\Rightarrow$] vector $\boldsymbol{P}^k$ of points and respective point levels $\boldsymbol{L}^k$ for each dimension $k \in [d]$



Spatially Adaptive SG CT & SparseSpACE Framework

Key components:

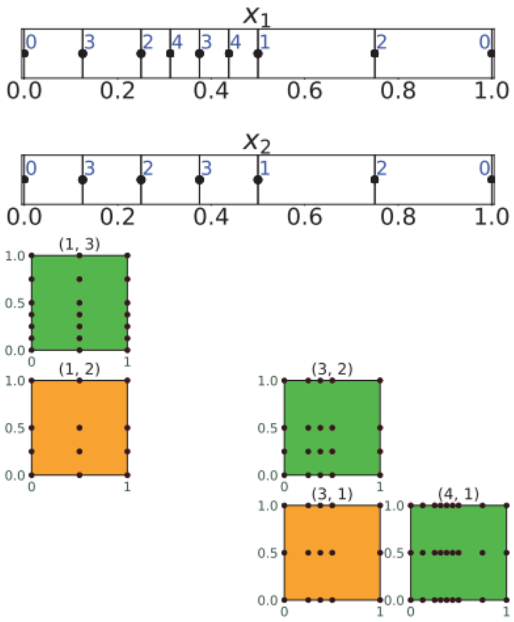
- Step 1: 1D refinements define the adaptive process
- Step 2: Creating a global valid combination scheme from 1D refinements
  - Step 2.1: create the index set based on the maximum levels $\boldsymbol{l}^{max}$ per dimension; where $l_k^{max} = max(\boldsymbol{L}^k)$

$$I = \{\boldsymbol{l} \in \mathbb{N}^d \mid \|\boldsymbol{l}\|_1 \leq max(\boldsymbol{l}^{max}) + d - 1, l_i \leq l_i^{max} \\ \vee (l_i = l_i^{max}, l_k = 1, k \in [d]/i)\} \qquad (2)$$

  - Step 2.2: define set of points $\boldsymbol{P}^{k,\boldsymbol{l}} \subseteq \boldsymbol{P}^k$ for each level vector $\boldsymbol{l}$ (i.e., $\boldsymbol{P}^k, \boldsymbol{L}^k \Rightarrow \boldsymbol{P}^{k,\boldsymbol{l}}, \boldsymbol{L}^{k,\boldsymbol{l}}$)
  - Ensure validity of combination scheme

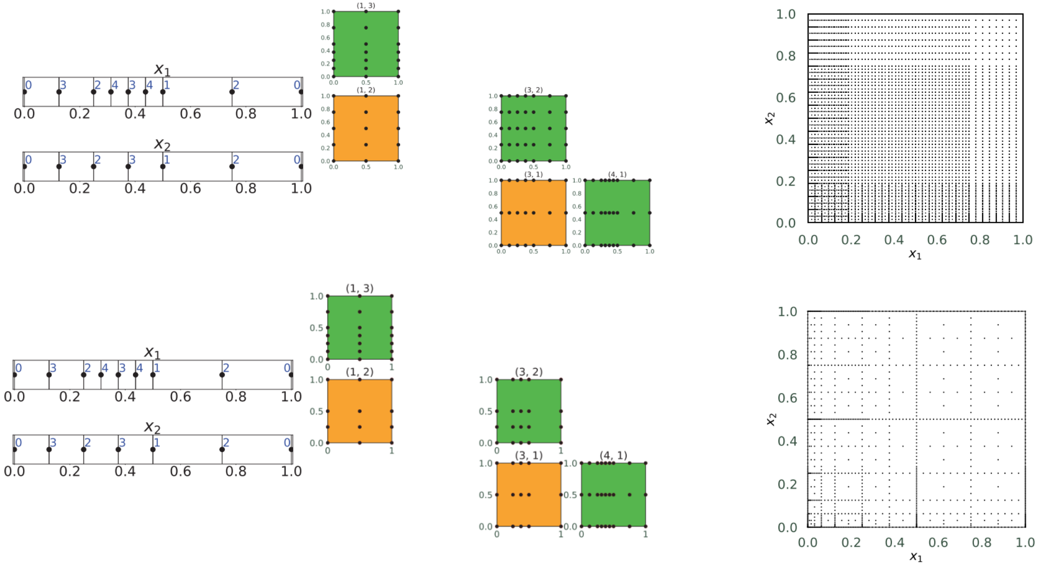
$$\boldsymbol{P}^{k,\boldsymbol{i}} \subseteq \boldsymbol{P}^{k,\boldsymbol{j}} \quad \text{for} \quad \boldsymbol{i}, \boldsymbol{j} \in I, k \in [d], \boldsymbol{j} \geq \boldsymbol{i} \qquad (3)$$

$$\boldsymbol{P}^{k,\boldsymbol{i}} = \boldsymbol{P}^{k,\boldsymbol{j}} \quad \text{for} \quad \boldsymbol{i}, \boldsymbol{j} \in I, k \in [d], i_k = j_k \qquad (4)$$



SparseSpACE Framework - creating component grids

Comparison of two strategies for creating component grids in SparseSpACE



Spatially Adaptive SG CT & SparseSpACE Framework

Complete algorithm:

- iterate over all component grids and calculate the 1D grid points for level vector $\boldsymbol{l} \in I$ (i.e., $\boldsymbol{P}^k, \boldsymbol{L}^k \Rightarrow \boldsymbol{P}^{k,\boldsymbol{l}}, \boldsymbol{L}^{k,\boldsymbol{l}}$)
- build via tensor construction the d-dimensional rectilinear grids
- **compute approximation** (e.g., interpolation or quadrature integration)
- compute error estimates for all leaf points
- refine every point $\varepsilon_p^{k,\boldsymbol{l}}$ with error estimate $|\varepsilon_p^k| \geq \gamma \cdot \varepsilon^{\max}$ (e.g., $\gamma = 0.5$)
- perform tree rebalancing for each dimension (update $\boldsymbol{P}^k, \boldsymbol{L}^k$)
- continue until tol. reached or max. num. of model evaluations

- **Example of the final approximation**

  (e.g., integration via quadrature approximation)

$$u_{\mathscr{I}}^{SCT} = \sum_{\boldsymbol{l} \in \mathscr{I}} c_{\boldsymbol{l}} \cdot \sum_{\boldsymbol{i} \in \prod_{k=1}^{d} [|\boldsymbol{P}^{k,\boldsymbol{l}}|]} \left(f(\theta^{\boldsymbol{i}}) \int_{\theta \in \Omega} \Psi_{\boldsymbol{i}}(\theta) d\theta \right) \quad (5)$$

  (e.g., interpolation)

$$u_I^{sct} = \sum_{\boldsymbol{l} \in I} c_{\boldsymbol{l}} \cdot \sum_{\boldsymbol{i} \in \prod_{k=1}^{d} [|\boldsymbol{P}^{k,\boldsymbol{l}}|]} \left(f(\theta^{\boldsymbol{i}}) \Psi_{\boldsymbol{i}}(\theta) \right) \quad (6)$$

- $\Psi_{\boldsymbol{i}}(\theta)$ are corresponding basis function assigned to point $\theta^{\boldsymbol{i}}$ with $\theta_k^{\boldsymbol{i}} = P_{i_k}^{k,\boldsymbol{l}}$

Non-intrusive Uncertainty Quantification

General Polynomial Chaos Expansion (gPCE) [8] of $f(t,\theta) : \mathbb{T} \times \Gamma \to \mathbb{R}$, reads:

$$f(t,\theta) \approx f_N(t,\theta) = \sum_{\boldsymbol{p}} c_{\boldsymbol{p}}(t)\Phi_{\boldsymbol{p}}(\theta) = \sum_{\boldsymbol{p}} < f(t,\theta), \Phi_{\boldsymbol{p}}(\theta) >_{\rho(\theta)} \Phi_{\boldsymbol{p}}(\theta) \tag{7}$$

- stochastic part - $\theta = (\theta_1, \theta_2, \dots, \theta_d)^T; \theta : \Omega \to \Gamma$ and $\rho(\theta) = \prod_{k=1}^{d} \rho_k(\theta_k)$
- $\boldsymbol{p} = (p_1, \dots, p_d)$ is a multi-index in $\mathscr{P}_P = \{\boldsymbol{p} \in \mathbb{N}^d : \sum_{k=1}^{d} p_k \leq P\}$,
- $\Phi_{\boldsymbol{p}}(\theta)$ are orthonormal multivariate polynomials constructed via a tensor product basis of the univariate polynomials $\Phi_{\boldsymbol{p}}(\theta) = \Phi_{p_1}(\theta_1) \cdot \ldots \cdot \Phi_{p_d}(\theta_d)$

Non-intrusive Uncertainty Quantification

General Polynomial Chaos Expansion (gPCE) [8] of $f(t,\theta) : \mathbb{T} \times \Gamma \to \mathbb{R}$, reads:

$$f(t,\theta) \approx f_N(t,\theta) = \sum_{\boldsymbol{p}} c_{\boldsymbol{p}}(t)\Phi_{\boldsymbol{p}}(\theta) = \sum_{\boldsymbol{p}} < f(t,\theta), \Phi_{\boldsymbol{p}}(\theta) >_{\rho(\theta)} \Phi_{\boldsymbol{p}}(\theta) \quad (7)$$

- stochastic part - $\theta = (\theta_1, \theta_2, \dots, \theta_d)^T; \theta : \Omega \to \Gamma$ and $\rho(\theta) = \prod_{k=1}^{d} \rho_k(\theta_k)$
- $\boldsymbol{p} = (p_1, \dots, p_d)$ is a multi-index in $\mathscr{P}_P = \{\boldsymbol{p} \in \mathbb{N}^d : \sum_{k=1}^{d} p_k \leq P\}$,
- $\Phi_{\boldsymbol{p}}(\theta)$ are orthonormal multivariate polynomials constructed via a tensor product basis of the univariate polynomials $\Phi_{\boldsymbol{p}}(\theta) = \Phi_{p_1}(\theta_1) \cdot \dots \cdot \Phi_{p_d}(\theta_d)$

Pseudo-spectral projection (PSP) - uses (full tensor) quadrature rule to approximate the coefficients of the gPCE

$$c_{\boldsymbol{p}}(t) = \mathbb{E}[f(t,\theta)\Phi_{\boldsymbol{p}}(\theta)] \approx \hat{c}_{\boldsymbol{p}}(t) = \sum_{\boldsymbol{q}=1}^{\boldsymbol{Q}} f(t,\theta^{\boldsymbol{q}})\Phi_{\boldsymbol{p}}(\theta^{\boldsymbol{q}})\omega^{\boldsymbol{q}} \quad (8)$$

Total number of coefficients: $N = \binom{P+d}{d}$

Total number of model evaluations: $\boldsymbol{Q} = \prod_{k=1}^{d} Q_k$; and it has to hold - $p_k = \text{floor}(DE(Q_k)/2)$ [9]

Non-intrusive Uncertainty Quantification
Post-processing & Sensitivity Analysis

Quantify uncertainty of f (or $\mathscr{O}(f)$) by computing, e.g.

$$E[f] = \int_\Gamma f(t,\theta)\rho(\theta)d\theta; \qquad Var[f] = E[f^2] - (E[f])^2 \tag{9}$$

Variance-based (Sobol) sensitivity analysis

$$S_k^T = \frac{Var(f) - Var(E(f|\theta_{-k}))}{Var(f)} = \frac{E(Var(f|\theta_{-k}))}{Var(f)} \tag{10}$$

Non-intrusive Uncertainty Quantification Post-processing & Sensitivity Analysis

Quantify uncertainty of f (or $\mathscr{O}(f)$) by computing, e.g.

$$E[f] = \int_{\Gamma} f(t,\theta)\rho(\theta)d\theta; \qquad Var[f] = E[f^2] - (E[f])^2 \tag{9}$$

Variance-based (Sobol) sensitivity analysis

$$S_k^T = \frac{Var(f) - Var(E(f|\theta_{-k}))}{Var(f)} = \frac{E(Var(f|\theta_{-k}))}{Var(f)} \tag{10}$$

Use gPCE coeff. to approximate expectation and variance:

$$\mathbb{E}[f_N(t,\theta)] = c_0(t) \qquad Var[f_N(t,\theta)] = \sum_{position(\boldsymbol{p})=1}^{N-1} c_{\boldsymbol{p}}^2(t) \tag{11}$$

Use gPCE coeff. to compute Sobol' indices (SI)[2]:

$$S_k^T = \frac{\sum_{\boldsymbol{p}\in A_k} c_{\boldsymbol{p}}^2(t)}{Var[f_N(t,\theta)]}, \quad A_k = \{\boldsymbol{p} \in \mathscr{P}_P \wedge \boldsymbol{p}_k \neq 0\} \tag{12}$$

UQ & Sparse Grids I

Multiple ways how to combine the gPCE (i.e., PSP) and SG.
So far SparseSpACE framework & UQ - adaptive SG integration quadrature rule used for computing integrals in $E[\mathscr{O}(f)]$ and $Var[\mathscr{O}(f)]$

Var 1: Sparse Quadrature (i.e., Sparse PSP)

Var 2: Sparse Interpolation Surrogate (i.e., f_{SGI}) + PSP

UQ & Sparse Grids II

Multiple ways how to combine the gPCE (i.e., PSP) and SG

Var 1: Sparse Quadrature (i.e., Sparse PSP)

$$
\begin{aligned}
\hat{c}_{n,I}(t) &= \sum_{\boldsymbol{l} \in I} c_{\boldsymbol{l}} \cdot S^{\boldsymbol{k}}_{\boldsymbol{l},n} \\
&= \sum_{\boldsymbol{l} \in I} c_{\boldsymbol{l}} \cdot \sum_{\boldsymbol{i} \in \prod_{k=1}^{d} [|\boldsymbol{P}^{k,\boldsymbol{l}}|]} \left(f(t, F^{-1}(\theta^{\boldsymbol{i}})) \Phi_n(F^{-1}(\theta^{\boldsymbol{i}})) \int_{\theta \in [0,1]^{\boldsymbol{d}}} \Psi_{\boldsymbol{i}}(\theta) d\theta \right) \\
&= \sum_{\boldsymbol{l} \in I} c_{\boldsymbol{l}} \cdot \sum_{\boldsymbol{i} \in \prod_{k=1}^{d} [|\boldsymbol{P}^{k,\boldsymbol{l}}|]} \left(f(t, F^{-1}(\theta^{\boldsymbol{i}})) \Phi_n(F^{-1}(\theta^{\boldsymbol{i}})) \omega^{\boldsymbol{i}} \right)
\end{aligned}
\tag{13}
$$

Problem with spatially adaptive approach - single adaptive SG integration rule needed for all the integrals $\hat{c}_{n,I}$

Var 2: Sparse Interpolation Surrogate (i.e., f_{SGI}) + PSP

where $\Psi_{\boldsymbol{i}}(\theta)$ are basis functions of the SG scheme, $\Phi_n(\theta)$ are basis polynomials of the PCE, $n \in [N]$ is a scalar index of the gPCE coeff. $\hat{c}_n$, and $c_{\boldsymbol{l}}$ is a scalar coeff. streaming from CT; $F^{-1} : [0,1]^{\boldsymbol{d}} \to \Gamma$ is an isoprobabilistic transformation of the variables in the probability space.

UQ & Sparse Grids III

Multiple ways how to combine the gPCE (i.e., PSP) and SG

Var 1: Sparse Quadrature (i.e., Sparse PSP)

Var 2: Sparse Interpolation Surrogate (i.e., f_{SGI}) + PSP

$$
\begin{aligned}
\hat{c}_{n,I}(t) &= \int_{\theta\in\Gamma} f_{\mathrm{SGI}}(t,\theta)\Phi_n(\theta)\rho(\theta)d\theta \\
&= \int_{\theta\in[0,1]^{\boldsymbol{d}}} \underbrace{\left(\sum_{\boldsymbol{l}\in I} c_{\boldsymbol{l}} \cdot \sum_{\boldsymbol{i}\in\prod_{k=1}^{d}[|\boldsymbol{P}^{k,\boldsymbol{l}}|]} f(t,F^{-1}(\theta^{\boldsymbol{i}}))\Psi_{\boldsymbol{i}}(\theta)\right)}_{f_{\mathrm{SGI}}} \Phi_n(F^{-1}(\theta))d\theta \\
&= \sum_{\boldsymbol{l}\in I} c_{\boldsymbol{l}} \cdot \sum_{\boldsymbol{i}\in\prod_{k=1}^{d}[|\boldsymbol{P}^{k,\boldsymbol{l}}|]} f(t,F^{-1}(\theta^{\boldsymbol{i}})) \int_{\theta\in[0,1]^{\boldsymbol{d}}} \Psi_{\boldsymbol{i}}(\theta)\Phi_n(F^{-1}(\theta))d\theta
\end{aligned}
\tag{14}
$$

where $\Psi_{\boldsymbol{i}}(\theta)$ are basis functions of the SG scheme, $\Phi_n(\theta)$ are basis polynomials of the PCE, $n\in[N]$ is a scalar index of the gPCE coeff. $\hat{c}_n$, and $c_{\boldsymbol{l}}$ is a scalar coeff. streaming from CT; $F^{-1}:[0,1]^{\boldsymbol{d}}\to\Gamma$ is an isoprobabilistic transformation of the variables in the probability space.

UQ & Sparse Grids - Initial Results

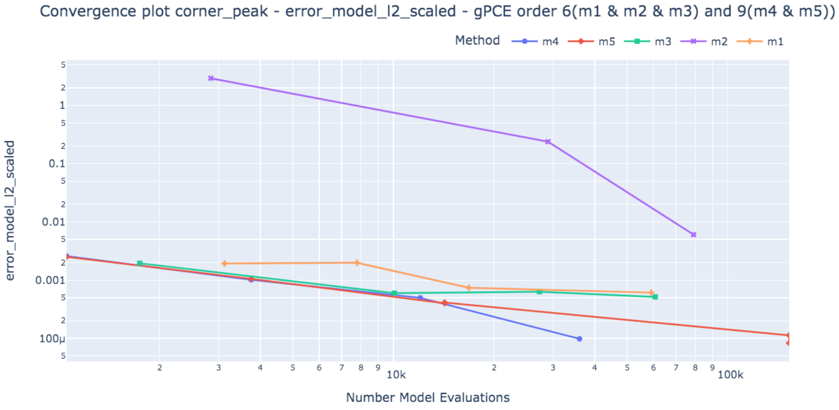
Variant	Method	Interpolation method (SGI)	quadrature method	gPCE
Var 1	m1	no	Full Gauss-Legendre	yes
	m2	no	(Sparse) Clenshaw-Curtis	yes
	m3	no	(Sparse) delayed Kronrod-Patterson[3]	yes
Var 2	m4	(piecewise linear) standard CT	Gauss-Legendre (high order) or analytical computation	yes
	m5	(piecewise linear) spatially adaptive CT	Gauss-Legendre (high order) or analytical computation	yes

Step 1: Benchmark Convergence of different methods

- **Surrogate construction:** SG-(gPCE) of **Genz function (5D)** and **Ishigami function (3D)**

$$f_{\text{corner}}(x) = \left(1 + \sum_{i=1}^{d} i \cdot x_i\right)^{-d-1} ; f_{\text{ishi}}(x) = sin(x_1) + a \cdot sin^2(x_2) + b \cdot x_3^4 \cdot sin(x_1)$$

- Practicalities:
  - linear basis functions
  - experiment with or without boundary points
  - using trapezoidal grid for sparse interpolation (m_4 & m_5)
  - refine up to 10% points with the largest surplus
- Convergence results as expected
- For simple cases, building an intermediate SG surrogate is not beneficial time-wise

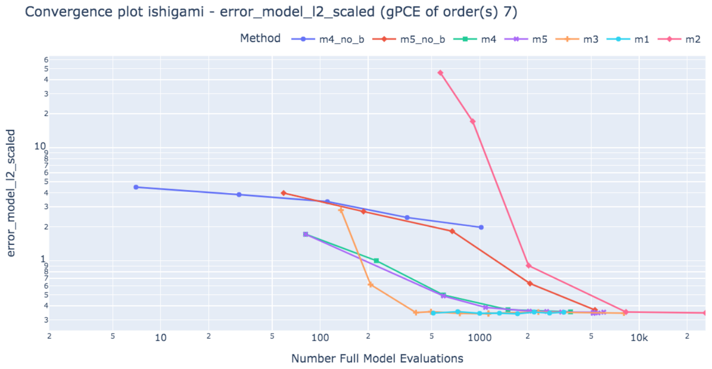


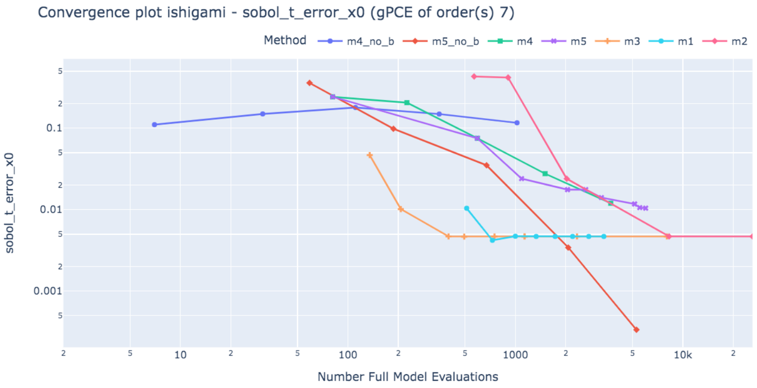
UQ & Sparse Grids - Initial Results

Variant	Method	Interpolation method (SGI)	quadrature method	gPCE
Var 1	m1	no	Full Gauss-Legendre	yes
	m2	no	(Sparse) Clenshaw-Curtis	yes
	m3	no	(Sparse) delayed Kronrod-Patterson[3]	yes
Var 2	m4	(piecewise linear) standard CT	Gauss-Legendre (high order) or analytical computation	yes
	m5	(piecewise linear) spatially adaptive CT	Gauss-Legendre (high order) or analytical computation	yes

Step 1: Benchmark Convergence of different methods

- **UQ & SA: Ishigami fun. (3D)** - analytical values for S.I. available

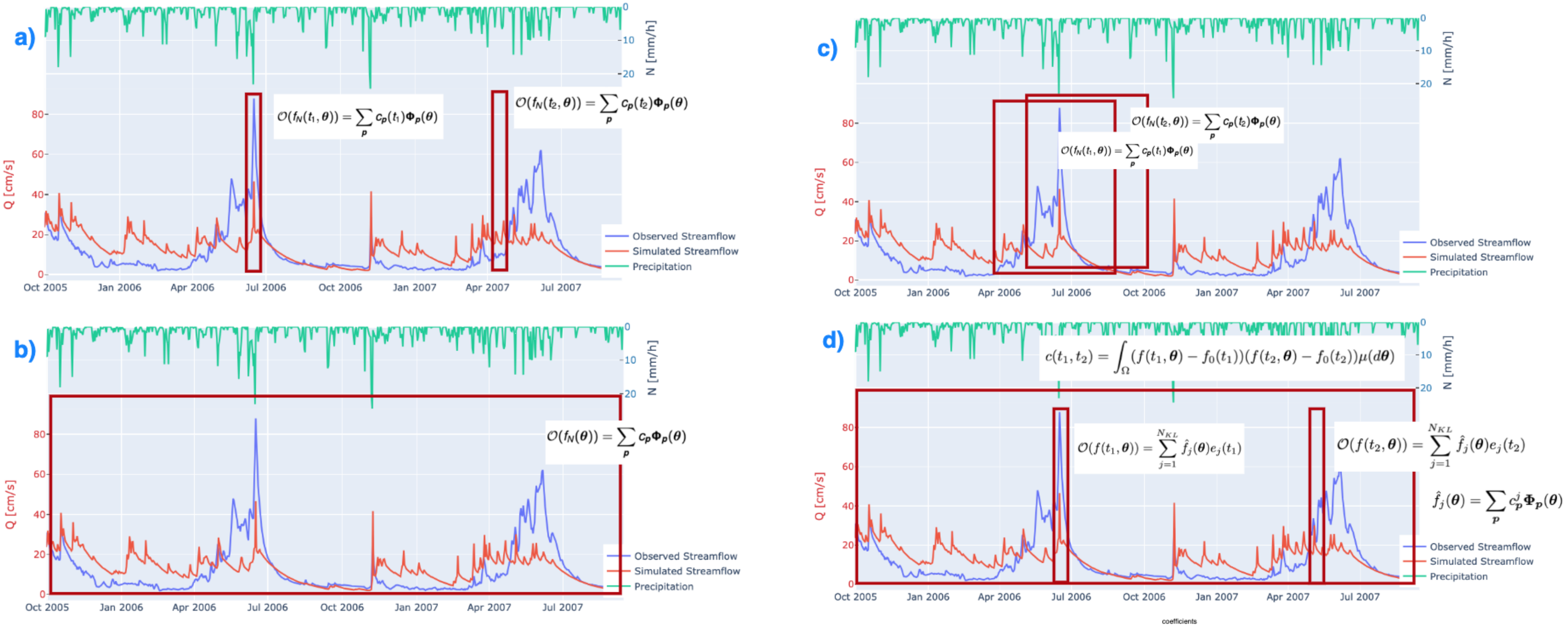




Time Dependent Analysis

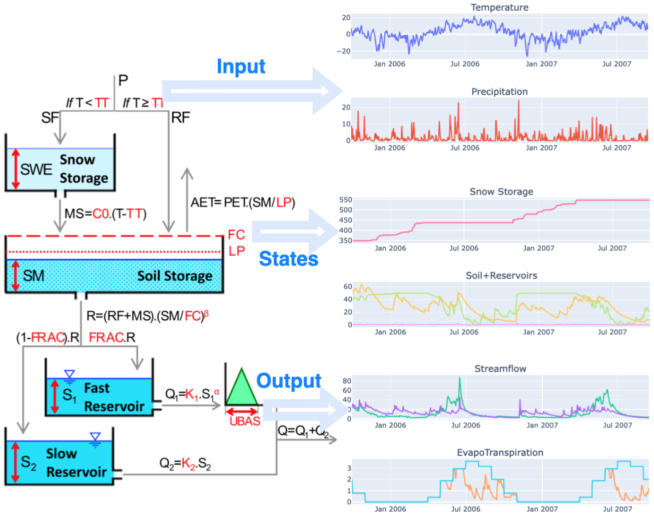
Different strategies: • a) Time-varying analysis; • b) Time-aggregated analysis; • c) Sliding-window analysis

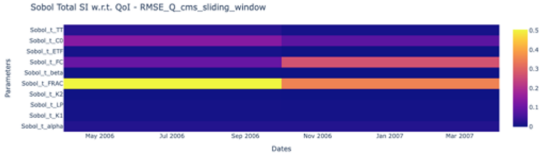
• d) Karhunen–Loéve (KL) expansion based intermediate surrogate

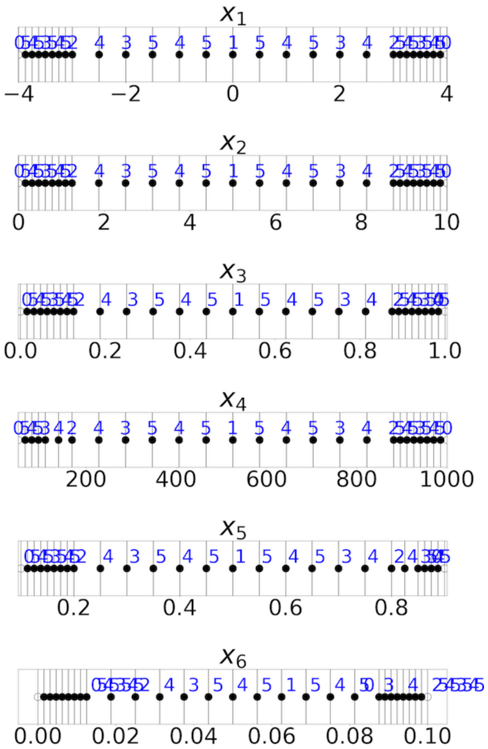


Time Aggregated UQ & SA with Adaptive SG

- **First Results - Time-aggregated:** Building a single (adpative) SG interpolation approximation of the data-misfit function (e.g., RMSE) for a certain time period;
- using it to learn gPCE surrogate of data-misfit function and derive Sobol S.I.
- convenient for identifying annual variability
- possible to use it as a surrogate model for efficient calibration instead of a full model







Future Work

- **SpareSpACE**
  - Introduce Parallelization in SpareSpACE (i.e., in parallel execution of a model for all the points in a single component grid)
  - Experiment with different grids/points (e.g., Leja) and basis functions (e.g., b-splines for interpolation)
- **Time-varying UQ & Sparse Grids**
  - Continue developing the KL intermediate surrogate and apply adaptive SG as needed

Thank You!

Bibliography

- Gupta, H. V. and Razavi, S., *Revisiting the Basis of Sensitivity Analysis for Dynamical Earth System Models*, Water Resources Research, 2018.
- Obersteiner, M. and Bungartz, H.-J., *A generalized spatially adaptive sparse grid combination technique with dimension-wise refinement*, SIAM Journal on Scientific Computing, 2021.
- Farcas, I. G., et al., *Revisiting the Basis of Sensitivity Analysis for Dynamical Earth System Models*, SIAM Journal on Scientific Computing, 2018.
- Bungartz, H.-J. and Griebel, M., *Sparse grids*, Acta Numerica, 2004.
- Griebel, M. and Schneider, M. and Zenger, C., *A combination technique for the solution of sparse grid problems*, Forschungsberichte, TU Munich, 1990.
- Gerstner, T. and Griebel, M., *Dimension–Adaptive Tensor–Product Quadrature*, Computing, 2003.
- Hegland, M., *Adaptive Sparse Grids*, ANZIAM J., 2001.
- Xiu, D and Karniadakis, G, *The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations*, SIAM Journal of Scientific Computing, 2002.
- Conrad, P. and Marzouk, Y., *Adaptive Smolyak Pseudosprectral Approximation*, , 2013.

Bibliography

- Constantine, P., et al. *Sparse pseudospectral approximation method*, Elsevier, 2012.
- Sudret, B., *Global sensitivity analysis using polynomial chaos expansions*, Reliability Engineering and System Safety, 2008.
- Knut, P., *Smolyak cubature of given polynomial degree with few nodes for increasing dimension*, Numerische Mathematik, 2003.

SparseSpACE Framework - creating component grids

Strategies for creating component grids out of 1D refinements:

- **Strategy 1** Add all points $p_j \in \boldsymbol{P}^k$ to $\boldsymbol{P}^{k,\boldsymbol{l}}$ for which $L_j^k \leq l_k$ for a component grid with level $\boldsymbol{l}$.

$$\boldsymbol{P}^{k,\boldsymbol{l}} = \{P_j^k \in \boldsymbol{P}^k \mid L_j^k \leq l_k\} \tag{15}$$

- **Strategy 2** Introduce the value c_j^k that should delay the level increase for dimension k.

$$\boldsymbol{P}^{k,\boldsymbol{l}} = \{P_j^k \in \boldsymbol{P}^k \mid L_j^k \leq l_k - c_j^k\}; \quad c_j^k = l_k^{\max} - D_j^k \tag{16}$$

  where D_j^k is the maximum of the levels of the hierarchical descendants of point P_j^k in the hierarchical tree.

- **Strategy 3** Control $0 \geq \hat{c}_j^k \leq c_j^k$; i.e., guarantees that leaves of $\boldsymbol{P}^k$ are added if we are at the level vector with the maximum level in dimension k