King Abdullah University of Science and Technology KAUST

Response Surface in low-rank Tensor Train Format for Uncertainty Quantification

Alexander Litvinenko[1]
(joint work with Sergey Dolgov[2], Boris Khoromsij[2] and Hermann G. Matthies[3])
[1] KAUST, [2] Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig,
[3] Technische Universität Braunschweig, Germany

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology | Center for Uncertainty Quantification

http://sri-uq.kaust.edu.sa/

Consider

$$A(u; q) = f \quad \Rightarrow \quad u = S(f; q),$$

where S is a solution operator.

Uncertain Input:

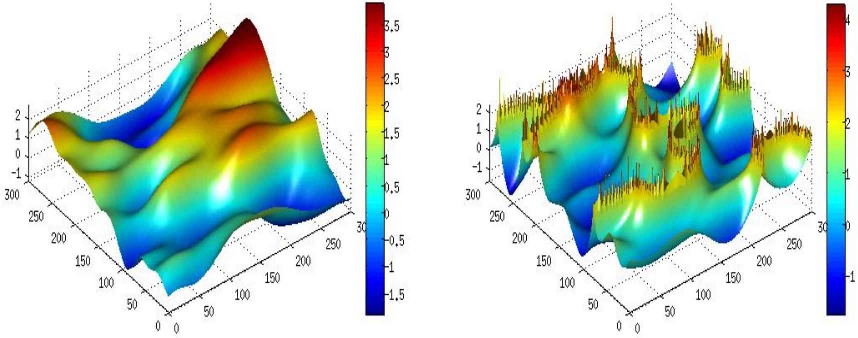
1. Parameter $q := q(\omega)$ (assume moments/cdf/pdf/quantiles of q are given)
2. Boundary and initial conditions, right-hand side
3. Geometry of the domain

Uncertain solution:

1. mean value and variance of u
2. exceedance probabilities $P(u > u^*)$
3. probability density functions (pdf) of u.

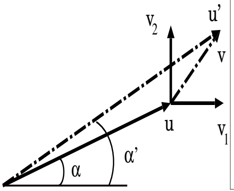
Nowadays computational algorithms, run on supercomputers, can simulate and resolve very complex phenomena. But how reliable are these predictions? Can we trust to these results?

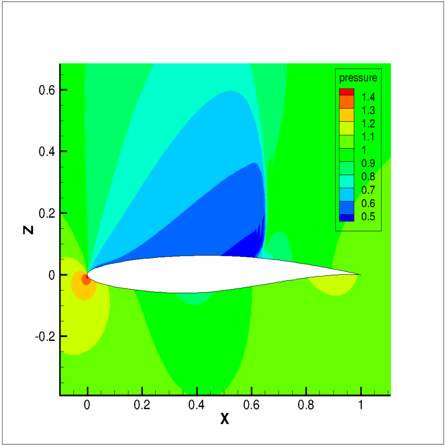
Some parameters/coefficients are unknown, lack of data, very few measurements $\rightarrow$ uncertainty.



2
1
0
-1
300
250
200
150
100
50
0
0
50
100
150
200
250
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
4
3
2
1
0
-1

What is Quantification of uncertainties ?
A big example:
UQ in numerical aerodynamics
(described by Navier-Stokes $+$ turbulence modeling)





Random vectors $\boldsymbol{v}_1(\theta)$ and $\boldsymbol{v}_2(\theta)$ model free stream turbulence

Example: UQ

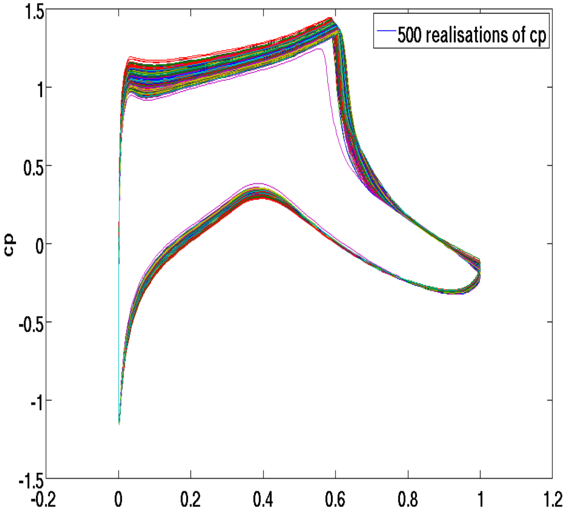
Input parameters: assume that RVs α and *Ma* are Gaussian with

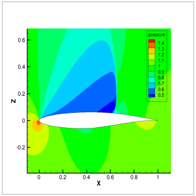
	mean	st. dev. σ	σ/mean
α	2.79	0.1	0.036
Ma	0.734	0.005	0.007

Then uncertainties in the solution (lift force and drag force) are

lift force	0.853	0.0174	0.02
drag force	0.0206	0.003	0.146

500 MC realisations of the pressure in dependence on α_i and Ma_i





$$\begin{cases} -\operatorname{div}(\kappa(x,\omega)\nabla u(x,\omega)) = p(x,\omega) & \text{in } \mathcal{G}\times\Omega,\ \mathcal{G}\subset\mathbb{R}^3, \\ u = 0 & \text{on } \partial\mathcal{G}, \end{cases} \tag{1}$$

where $\kappa(x,\omega)$ - conductivity coefficient. Since κ positive, usually $\kappa(x,\omega) = e^{\gamma(x,\omega)}$.

Discretisation of stochastic PDE

The Karhunen-Loève expansion is the series

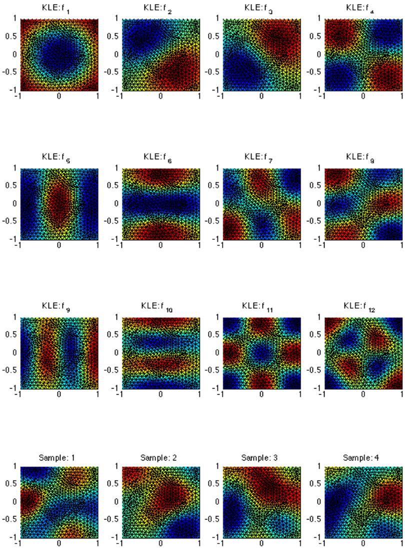
$$\kappa(x,\omega) = \mu_k(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} k_i(x) \xi_i(\omega), \quad \text{where}$$

$\xi_i(\omega)$ are uncorrelated random variables and k_i are basis functions in $L^2(\mathcal{G})$.
Eigenpairs λ_i, k_i are the solution of

$$T k_i = \lambda_i k_i, \quad k_i \in L^2(\mathcal{G}), i \in \mathbb{N}, \quad \text{where.}$$

$$T : L^2(\mathcal{G}) \to L^2(\mathcal{G}),$$
$$(Tu)(x) := \int_{\mathcal{G}} \operatorname{cov}_k(x,y) u(y) dy.$$

KLE eigenfunctions in 2D



$$\xi_i(\omega) \approx \sum_{k=0}^{Z} a_k \Psi_k(\theta_1, \theta_2, ..., \theta_M),$$

where $Z = \frac{(M+p)!}{M!p!}$ or $Z = p^M$:
- EXPENSIVE!
$M = 9$, $p = 2$, $Z = 55$
$M = 9$, $p = 4$, $Z = 715$
$M = 100$, $p = 4$, $Z \approx 4 \cdot 10^6$.
How to store and to handle so many coefficients ?
The orthogonality of Ψ_k enables the evaluation

$$a_k = \frac{< \xi \Psi_k >}{< \Psi_k^2 >} = \frac{1}{< \Psi_k^2 >} \int \xi(\boldsymbol{\theta}(\omega))\Psi_k(\boldsymbol{\theta}(\omega))dP(\omega).$$

(e.g. Ψ_k are multivariate Hermite polynomials).

We assume $\kappa = \phi(\gamma)$ -a smooth transformation of the Gaussian random field $\gamma(x, \omega)$, e.g. $\phi(\gamma) = \exp(\gamma)$.
Expanding ϕ in a series in the Hermite polynomials:

$$\phi(\gamma) = \sum_{i=0}^{\infty} \phi_i h_i(\gamma), \quad \phi_i = \int_{-\infty}^{+\infty} \phi(z) \frac{1}{i!} h_i(z) \exp(-z^2/2) dz, \quad (2)$$

where $h_i(z)$ is the i-th Hermite polynomial.
[see PhD of E. Zander 2013, or PhD of A. Keese, 2005]

First, given the covariance matrix of $\kappa(x,\omega)$, we may relate it with the covariance matrix of $\gamma(x,\omega)$ as follows,

$$\mathrm{cov}_\kappa(x,y) = \int (\kappa(x,\omega) - \bar{\kappa}(x))\,(\kappa(y,\omega) - \bar{\kappa}(y))\,dP(\omega)$$
$$\approx \sum_{i=0}^{Q} i!\,\phi_i^2\,\mathrm{cov}_\gamma^i(x,y).$$

Solving this implicit Q-order equation [E. Zander, 13], we derive $\mathrm{cov}_\gamma(x,y)$. Now, the KLE may be computed,

$$\gamma(x,\omega) = \sum_{m=1}^{\infty} g_m(x)\theta_m(\omega), \quad \int_D \mathrm{cov}_\gamma(x,y) g_m(y) dy = \lambda_m g_m(x), \tag{3}$$

Full $\mathcal{J}_{M,p}$ and sparse $\mathcal{J}^{sp}_{M,p}$ multi-index sets

Definition

The *full* multi-index is defined by restricting each component independently,

$$\mathcal{J}_{M,p} = \{0, 1, \dots, p_1\} \otimes \cdots \otimes \{0, 1, \dots, p_M\}, \quad \text{where} \quad p = (p_1, \dots, p_M)$$

is a shortcut for the tuple of order limits.

Definition

The *sparse* multi-index is defined by restricting the sum of components,

$$\mathcal{J}^{sp}_{M,p} = \{\alpha = (\alpha_1, \dots, \alpha_M) : \quad \alpha \geq 0, \ \ \alpha_1 + \cdots + \alpha_M \leq p\}.$$

As a result, the M-dimensional PCE approximation of κ writes

$$\kappa(x,\omega) \approx \sum_{\alpha \in \mathcal{J}_M} \kappa_\alpha(x) H_\alpha(\theta(\omega)), \quad H_\alpha(\theta) := h_{\alpha_1}(\theta_1)\cdots h_{\alpha_M}(\theta_M) \tag{4}$$

The Galerkin coefficients κ_α are evaluated as follows [Thm 3.10, PhD of E. Zander 13],

$$\kappa_\alpha(x) = \frac{(\alpha_1 + \cdots + \alpha_M)!}{\alpha_1! \cdots \alpha_M!} \phi_{\alpha_1+\cdots+\alpha_M} \prod_{m=1}^{M} g_m^{\alpha_m}(x), \tag{5}$$

where $\phi_{|\alpha|} := \phi_{\alpha_1+\cdots+\alpha_M}$ is the Galerkin coefficient of the transform function in (2), and $g_m^{\alpha_m}(x)$ means just the α_m-th power of the KLE function value $g_m(x)$.

Complexity reduction in Eq. (5) can be achieved with the help of the KLE for the initial field $\kappa(x,\omega)$:

$$\kappa(x,\omega) = \bar{\kappa}(x) + \sum_{\ell=1}^{\infty} \sqrt{\mu_\ell} v_\ell(x) \eta_\ell(\omega) \tag{6}$$

with the normalized spatial functions $v_\ell(x)$.
Instead of using (5) directly, we compute

$$\tilde{\kappa}_\alpha(\ell) = \frac{(\alpha_1 + \cdots + \alpha_M)!}{\alpha_1! \cdots \alpha_M!} \phi_{\alpha_1+\cdots+\alpha_M} \int_D \prod_{m=1}^{M} g_m^{\alpha_m}(x) v_\ell(x) dx. \tag{7}$$

Note that $L \ll N$. Then we restore the approximate coefficients

$$\kappa_\alpha(x) \approx \bar{\kappa}(x) + \sum_{\ell=1}^{L} v_\ell(x) \tilde{\kappa}_\alpha(\ell). \tag{8}$$

Given Eq.6, assemble

$$K_0(i,j) = \int_D \bar{\kappa}(x)\nabla\varphi_i(x)\cdot\nabla\varphi_j(x)dx, \;\; K_\ell(i,j) = \int_D v_\ell(x)\nabla\varphi_i(x)\cdot\nabla\varphi_j(x)d \tag{9}$$

for $i,j = 1,\dots,N$, $\ell = 1,\dots,L$. Take $\tilde{\kappa}_\alpha(\ell)$ and integrate over θ:

$$K_{\alpha,\beta}(\ell) = \int_{\mathbb{R}^M} H_\alpha(\theta)H_\beta(\theta) \sum_{\gamma\in\mathcal{J}_{M,p}} \kappa_\gamma(\ell)H_\gamma(\theta)d\theta = \sum_{\gamma\in\mathcal{J}_{M,p}} \boldsymbol{\Delta}_{\alpha,\beta,\gamma}\kappa_\gamma(\ell), \tag{10}$$

where

$$\boldsymbol{\Delta}_{\alpha,\beta,\gamma} = \Delta_{\alpha_1,\beta_1,\gamma_1}\cdots\Delta_{\alpha_M,\beta_M,\gamma_M}, \tag{11}$$

$$\Delta_{\alpha_m,\beta_m,\gamma_m} = \int_{\mathbb{R}} h_{\alpha_m}(z)h_{\beta_m}(z)h_{\gamma_m}(z)dz, \tag{12}$$

is the triple product of the Hermite polynomials.

Putting together (8), (9) and (10), we obtain the whole discrete stochastic Galerkin operator,

$$\mathbf{K} = K_0 \otimes \mathbf{\Delta}_0 + \sum_{\ell=1}^{L} K_\ell \otimes \sum_{\gamma \in \mathcal{J}_{M,p}} \mathbf{\Delta}_\gamma \tilde{\kappa}_\gamma(\ell), \qquad (13)$$

which $\mathbf{K} \in \mathbb{R}^{N(p+1)^M \times N(p+1)^M}$ in case of full $\mathcal{J}_{M,p}$.
If $\tilde{\kappa}_\gamma$ is computed in the tensor product format, the direct product in $\mathbf{\Delta}$ (11) allows to exploit the same format for (13), and build the operator easily.

Two tensor Train examples

$$f(x_1,...,x_d) = w_1(x_1) + w_2(x_2) + ... + w_d(x_d)$$

$$= (w_1(x_1), 1) \begin{pmatrix} 1 & 0 \\ w_2(x_2) & 1 \end{pmatrix} ... \begin{pmatrix} 1 & 0 \\ w_{d-1}(x_{d-1}) & 1 \end{pmatrix} \begin{pmatrix} 1 \\ w_d(x_d) \end{pmatrix}$$

Examples:

rank(f)=2

$$f = \sin(x_1 + x_2 + ... + x_d)$$

$$= (\sin x_1, \cos x_1) \begin{pmatrix} \cos x_2 & -\sin x_2 \\ \sin x_2 & \cos x_2 \end{pmatrix} \dots \begin{pmatrix} \cos x_{d-1} & -\sin x_{d-1} \\ \sin x_{d-1} & \cos x_{d-1} \end{pmatrix} \begin{pmatrix} \cos x_d \\ \sin x_{d-1} \end{pmatrix}$$

Tensor Train decomposition

$$u(\alpha) = \tau(u^{(1)}, \dots, u^{(M)}), \quad \text{meaning}$$

$$u(\alpha_1, \dots, \alpha_M) = \sum_{s_1=1}^{r_1} \sum_{s_2=1}^{r_2} \cdots \sum_{s_{M-1}=1}^{r_{M-1}} u^{(1)}_{s_1}(\alpha_1) u^{(2)}_{s_1,s_2}(\alpha_2) \cdots u^{(M)}_{s_{M-1}}(\alpha_M),$$

$$u(\alpha_1, \dots, \alpha_M) = u^{(1)}(\alpha_1) u^{(2)}(\alpha_2) \cdots u^{(M)}(\alpha_M), \quad \text{or}$$

$$u = \sum_{s_1=1}^{r_1} \sum_{s_2=1}^{r_2} \cdots \sum_{s_{M-1}=1}^{r_{M-1}} u^{(1)}_{s_1} \otimes u^{(2)}_{s_1,s_2} \otimes \cdots \otimes u^{(M)}_{s_{M-1}}. \tag{14}$$

Each TT core $u^{(k)} = [u^{(k)}_{s_{k-1},s_k}(\alpha_k)]$ is defined by $r_{k-1} n_k r_k$ numbers, where n_k is number of grid points (e.g. $n_k = p_k + 1$) in the α_k direction, and r_k is the *TT rank*. The total number of entries $\mathcal{O}(Mnr^2)$, $r = \max\{r_k\}$.

It has the Kronecker (canonical) rank-M representation:

$$\mathbf{A} = A \otimes I \otimes \cdots \otimes I + I \otimes A \otimes \cdots \otimes I + \cdots + I \otimes I \otimes \cdots \otimes A \in \mathbb{R}^{n^M \times n^M} \quad (15)$$

with $A = \text{tridiag}\{-1, 2, -1\} \in \mathbb{R}^{n \times n}$, and I the $n \times n$ identity.
In the TT format is explicitly representable with all TT ranks equal to 2:

$$\mathbf{A} = (A \ I) \bowtie \begin{pmatrix} I & 0 \\ A & I \end{pmatrix} \bowtie \ldots \bowtie \begin{pmatrix} I & 0 \\ A & I \end{pmatrix} \bowtie \begin{pmatrix} I \\ A \end{pmatrix}, \quad (16)$$

Or

$$\mathbf{A}(\mathbf{i}, \mathbf{j}) = \begin{pmatrix} A(i_1, j_1) & I(i_1, j_1) \end{pmatrix} \begin{pmatrix} I(i_2, j_2) & 0 \\ A(i_2, j_2) & I(i_2, j_2) \end{pmatrix} \cdots \begin{pmatrix} I(i_d, j_d) \\ A(i_d, j_d) \end{pmatrix}.$$

Calculation of

$$\tilde{\kappa}_{\alpha}(\ell) = \frac{(\alpha_1 + \cdots + \alpha_M)!}{\alpha_1! \cdots \alpha_M!} \phi_{\alpha_1 + \cdots + \alpha_M} \int_D \prod_{m=1}^{M} g_m^{\alpha_m}(x) v_\ell(x) dx.$$

in tensor formats needs:

- given a procedure to compute each element of a tensor, e.g. $\tilde{\kappa}_{\alpha_1,\ldots,\alpha_M}$ by (26).
- build a TT approximation $\tilde{\kappa}_\alpha \approx \kappa^{(1)}(\alpha_1) \cdots \kappa^{(M)}(\alpha^M)$ using a feasible amount of elements (i.e. much less than $(p+1)^M$).

Such procedure exists, and relies on the *cross interpolation* of matrices, generalized to a higher-dimensional case [Oseledets, Tyrtyshnikov 2010; Savostyanov 13; Grasedyck; Bebendorf].

Skip 3 technical slides about Maximum volume principle and its application

As soon as the reduced PCE coefficients $\tilde{\kappa}_\alpha(\ell)$ are computed, the initial expansion (8) comes easily. Indeed, stop the cross iteration at the first block, that is

$$\tilde{\kappa}_\alpha(\ell) = \sum_{s_1,\ldots,s_{M-1}} \kappa^{(1)}_{\ell,s_1}(\alpha_1)\cdots\kappa^{(M)}_{s_{M-1}}(\alpha_M). \tag{17}$$

Now, collect the spatial components into the “zeroth” TT block,

$$\kappa^{(0)}(x) = \left[\kappa^{(0)}_\ell(x)\right]_{\ell=0}^{L} = \left[\bar{\kappa}(x) \quad v_1(x) \quad \cdots v_L(x)\right], \tag{18}$$

then the PCE (4) writes as the following TT format,

$$\kappa_\alpha(x) = \sum_{\ell,s_1,\ldots,s_{M-1}} \kappa^{(0)}_\ell(x)\kappa^{(1)}_{\ell,s_1}(\alpha_1)\cdots\kappa^{(M)}_{s_{M-1}}(\alpha_M). \tag{19}$$

Given (19), we split the whole sum over γ in (13):

$$\sum_{\gamma\in\mathcal{J}_{M,p}} \mathbf{\Delta}_\gamma \tilde{\kappa}_\gamma(\ell) = \sum_{s_1,\ldots,s_{M-1}} \left(\sum_{\gamma_1=0}^{p} \Delta_{\gamma_1} \kappa^{(1)}_{\ell,s_1}(\gamma_1)\right) \otimes\cdots\otimes \left(\sum_{\gamma_M=0}^{p} \Delta_{\gamma_M} \kappa^{(M)}_{s_{M-1}}(\gamma_M)\right).$$

Introduce

$$\mathbf{K}^{(0)}(i,j) := \left[\mathbf{K}^{(0)}_\ell(i,j)\right]_{\ell=0}^{L} = \left[K_0(i,j) \quad K_1(i,j) \quad \cdots \quad K_L(i,j)\right], \; i,j = 1,\ldots,N,$$

$\mathbf{K}^{(m)}_{s_{m-1},s_m} := \sum_{\gamma_m=0}^{p} \Delta_{\gamma_m} \kappa^{(m)}_{s_{m-1},s_m}(\gamma_m)$ for $m = 1,\ldots,M$,

then the TT representation for the operator writes

$$\mathbf{K} = \sum_{\ell,s_1,\ldots,s_{M-1}} \mathbf{K}^{(0)}_\ell \otimes \mathbf{K}^{(1)}_{\ell,s_1} \otimes\cdots\otimes \mathbf{K}^{(M)}_{s_{M-1}} \in \mathbb{R}^{(N\cdot\#\mathcal{J}_{M,p})\times(N\cdot\#\mathcal{J}_{M,p})}, \quad (20)$$

Post-processing:

We compute
Characteristic, level sets, frequency in TT format

Numerics: Main steps

1. Compute PCE of the coefficients $\kappa(x, \omega)$ in TT format
2. Compute stochastic Galerkin matrix $\mathbf{K}$ in TT
3. Compute solution of the linear system in TT
4. Post-processing in TT format

$\kappa(x,\omega)$ obeys the $\beta\{5,2\}$-distribution,
$\mathrm{cov}_\kappa(x,y) = \exp\left(-(x-y)^2/\sigma^2\right)$ with $\sigma = 0.3$. D is L-shape domain, 557 DOFs.
Use *sglib* (E. Zander, TU BS) for discretization and solution with $\mathcal{J}^{sp}_{M,p}$.
Use *TT-Toolbox* for full $\mathcal{J}_{M,p}$.
Use *sglib* for low-dimensional stages,
and replace high-dimensional calculations by the TT.
Use `amen_cross.m` for TT approximation of $\tilde{\kappa}_\alpha$ (26),
Use `amen_solve.m` (`tAMEn`, Dolgov) as linear system solver in TT format.

Table : CPU times (sec.) of the permeability assembly

	Sparse			TT		
$p \setminus M$	10	20	30	10	20	30
1	0.2924	0.3113	0.3361	3.6425	68.505	616.97
2	0.3048	0.3556	0.4290	6.3861	138.31	1372.9
3	0.3300	0.5408	1.0302	8.8109	228.92	2422.9
4	0.4471	1.7941	6.4483	10.985	321.93	3533.4
5	1.1291	7.6827	46.682	14.077	429.99	4936.8

Table : Discrepancies in the permeability coefficients at $\mathcal{J}^{sp}_{M,p}$

p	1	2	3	4	5
$M = 10$	2.21e-4	3.28e-5	1.22e-5	4.15e-5	6.38e-5
$M = 20$	3.39e-4	5.19e-5	2.20e-5	—	—
$M = 30$	5.23e-2	5.34e-2	—	—	—

CPU times (sec.) of the operator assembly

	Sparse			TT		
$p \setminus M$	10	20	30	10	20	30
1	0.1226	0.2171	0.3042	0.1124	0.2147	0.3836
2	0.1485	2.1737	26.510	0.1116	0.2284	0.5438
3	2.2483	735.15	—	0.1226	0.2729	0.8403
4	82.402	—	—	0.1277	0.2826	1.0832
5	3444.6	—	—	0.2002	0.3495	1.1834

CPU times (sec.) of the solution

	Sparse			TT		
$p \setminus M$	10	20	30	10	20	30
1	0.2291	1.169	0.4778	1.074	9.3492	51.177
2	0.3088	2.123	3.2153	1.681	27.014	173.21
3	0.8112	14.04	—	2.731	56.041	391.59
4	5.7854	—	—	7.237	142.87	1497.1
5	61.596	—	—	45.51	866.07	5362.8

The reference covariance matrix $\mathrm{cov}_u^\star \in \mathbb{R}^{N\times N}$ is computed in the TT format with $p = 5$, and the discrepancies in the results with smaller p are calculated in average over all spatial points,

$$|\,\mathrm{cov}_u - \mathrm{cov}_u^\star\,| = \frac{\sqrt{\sum_{i,j}(\mathrm{cov}_u - \mathrm{cov}_u^\star)^2_{i,j}}}{\sqrt{\sum_{i,j}(\mathrm{cov}_u^\star)^2_{i,j}}}.$$

	Sparse			TT		
$p \setminus M$	10	20	30	10	20	30
1	9.49e-2	8.86e-2	9.67e-2	4.18e-2	2.80e-2	2.60e-2
2	3.46e-3	2.65e-3	3.34e-3	1.00e-4	1.31e-4	2.12e-4
3	1.65e-4	2.77e-4	—	4.48e-5	1.32e-4	2.14e-4
4	8.58e-5	—	—	6.28e-5	1.33e-4	1.11e-4

1. demonstrated RS in TT format for solving PDEs with uncertain coefficients.
2. Favor of the TT comparing to CP is a stable quasi-optimal rank reduction based on SVD.
3. Complexity $\mathcal{O}(Mnr^3)$ with full accuracy control.
4. TT methods become preferable for high p, but otherwise the full computation in a small sparse set may be incredibly fast. This reflects well the "curse of order", taking place for the sparse set instead of the "curse of dimensionality" in the full set: the cardinality of the sparse set grows exponentially with p.
5. The TT approach scales linearly with p.

Take to home

1. TT methods allow easy calculation of the stochastic Galerkin operator. With p below 10, the TT storage of the operator allows us to forget about the sparsity issues, since the number of TT entries $\mathcal{O}(Mp^2 r^2)$ is tractable.
2. Other polynomial families, such as the Chebyshev or Laguerre, may be incorporated into the scheme freely.
3. TT formalism may be recommended for stochastic PDEs as a general tool: one introduces the same discretization levels for all variables and let the algorithms determine a quasi-optimal representation adaptivity.

Many questions are still open

1. Can we endow the solution scheme with more structure and obtain a more efficient algorithm?
2. Is there a better way to discretize stochastic fields than the KLE-PCE approach?
3. In the preliminary experiments, we have investigated only the simplest statistics, i.e. mean and variance. What quantities (level sets, frequency,...) are feasible in TT format and how can they be effectively computed?

Stochastic Galerkin library

1. Type in your terminal
   `git clone git://github.com/ezander/sglib.git`
2. To initialize all variables, run `startup.m`

You will find:
generalised PCE, sparse grids, (Q)MC, stochastic Galerkin, linear solvers, KLE, covariance matrices, statistics, quadratures (multivariate Chebyshev, Laguerre, Lagrange, Hermite) etc

There are: many examples, many test, rich demos