ÚGN

Introduction to the stochastic Galerkin approach for the solution of PDEs with parameters or uncertainties

Michal Béreš

30th October 2019

Outline

Recapitulation of Random variable and Probability measure

Structure of the problem, discretization and suitable problems

System of equations and assembly

Approaches to the solution of the system

Toy example, introduction to the problem

- 2D Darcy flow problem

$$\begin{cases} -\mathrm{div}\left(k\left(x;\boldsymbol{Z}\right)\cdot\nabla u\left(x;\boldsymbol{Z}\right)\right)=f\left(x;\boldsymbol{Z}\right), & \forall x\in\mathcal{D} \\ u\left(x;\boldsymbol{Z}\right)=u_D\left(x;\boldsymbol{Z}\right), & \forall x\in\Gamma_D\ , \\ \frac{\partial u(x;\boldsymbol{Z})}{\partial n(x)}=u_N\left(x;\boldsymbol{Z}\right), & \forall x\in\Gamma_N \end{cases}$$

$\boldsymbol{Z}$ can be understood as $\boldsymbol{Z}:\Omega\to\mathbb{R}^M$ and $k\left(x;\boldsymbol{Z}\right)$,
$u\left(x;\boldsymbol{Z}\right):\mathcal{D}\times\mathbb{R}^M\to\mathbb{R}$

Motivation, intrusive and non-intrusive approaches

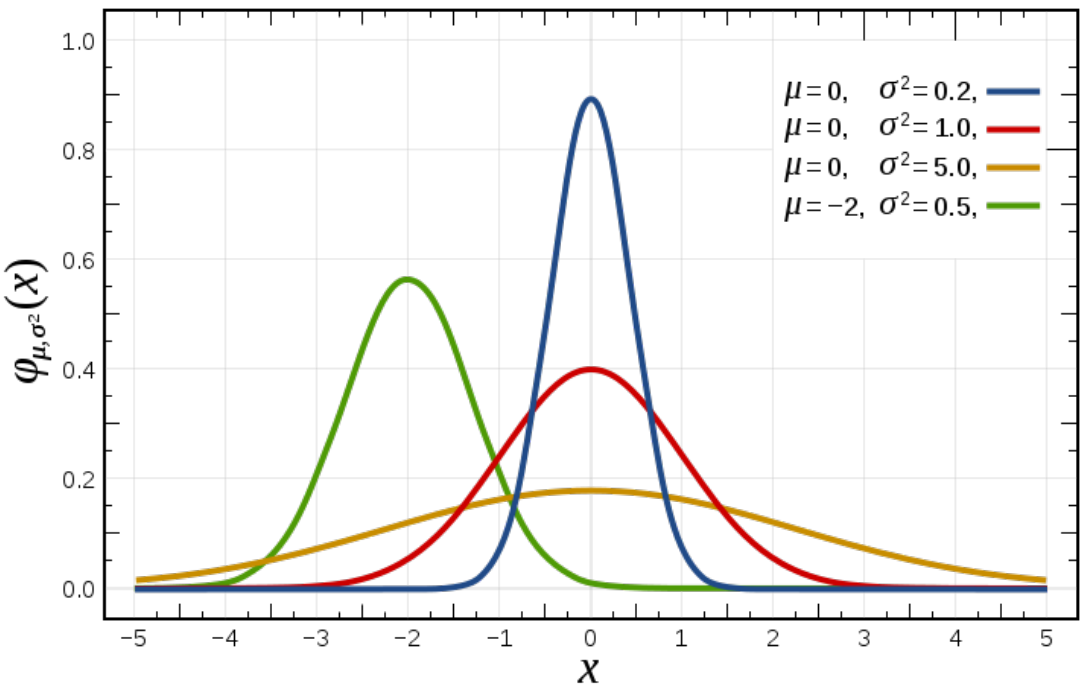
- **non-intrusive** approaches examines only point values of the solution (much easier to deploy)
  - PDE solution in fixed values of the parameter is handled as **black-box** by an existing software
  - Collocation or Monte Carlo methods
- stochastic Galerkin (SG) method is an **intrusive** approach
  - needs new framework for the solution of given problems
  - it provides error estimates

Recapitulation of Random variable and Probability measure

Random variables and probability measure

- continuous random variable Z is a map from sample space $\Omega \to \mathbb{R}$
- can be described by probability density $f : \mathbb{R} \to \mathbb{R}_0^+$, $\int_{\mathbb{R}} f(x)\,\mathrm{d}x = 1$
- distribution of a random variable defines a probability measure
  - in the case of a continuous random variable

$$\int_{\mathbb{R}} \ldots \mathrm{d}FZ = \int_{\mathbb{R}} \ldots f(x)\,\mathrm{d}x = \mathbb{E}_Z(\ldots)$$



Random vector

- is a vector of M (let assume continuous) random variables $\boldsymbol{Z} = (Z_1, \ldots, Z_M)$
- can be described by joint probability density $f_{\boldsymbol{Z}} : \mathbb{R}^M \to \mathbb{R}_0^+$, $\int\limits_{\mathbb{R}^M} f(x)\,\mathrm{d}x = 1$
- random vector of independent random variables has joint probability density in form

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \prod_{i=1}^{M} f_{Z_i}(z_i)$$

L^2 spaces defined by a random vector

- square integrable functions of a random vector $\boldsymbol{Z}$ creates space

$$L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M\right) := \left\{ f : \mathbb{R}^M \to \mathbb{R} : \int_{\mathbb{R}^M} f\left(\boldsymbol{z}\right)^2 dF\boldsymbol{Z} < \infty \right\}$$

- with inner product

$$\left(u, v\right) := \int_{\mathbb{R}^M} u\left(\boldsymbol{z}\right) \cdot v\left(\boldsymbol{z}\right) dF\boldsymbol{Z} = \mathbb{E}_{\boldsymbol{Z}}\left(u\left(\boldsymbol{Z}\right) \cdot v\left(\boldsymbol{Z}\right)\right)$$

Polynomial chaos in 1d

- **orthogonal polynomial basis of $L^2_{\mathrm{d}FZ}$**
- polynomial chaos with respect to a random variable Z
- set of polynomials of increasing degree $\psi_i(Z)$ (i denotes degree of polynomial)
- orthogonal $=$ satisfying $\mathbb{E}_Z(\psi_i(Z)\psi_j(Z)) = \gamma_i\delta_{ij}$, where $\gamma_i = \mathbb{E}_Z\left(\psi_i(Z)^2\right)$
- generates basis of $L^2_{\mathrm{d}FZ}(\mathbb{R})$

Polynomial chaos Nd

- assume random vector $\boldsymbol{Z}$ of **independent** random variables
- polynomial chaos for $L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M\right)$ can be created by a product of 1d polynomials of Z_i

$$\Psi_i\left(\boldsymbol{Z}\right) = \prod_{k=1}^{M} \psi_{i_k}\left(Z_k\right),$$

  where i denotes the multi-index of size M.

tensor product polynomials of maximal degree n are polynomials with

$$i_k \leq n, \forall k = 1, \ldots, M$$

complete polynomials of maximal degree n are polynomials with

$$|i| = \sum_{k=1}^{M} i_k \leq n$$

Complete vs. tensor polynomials

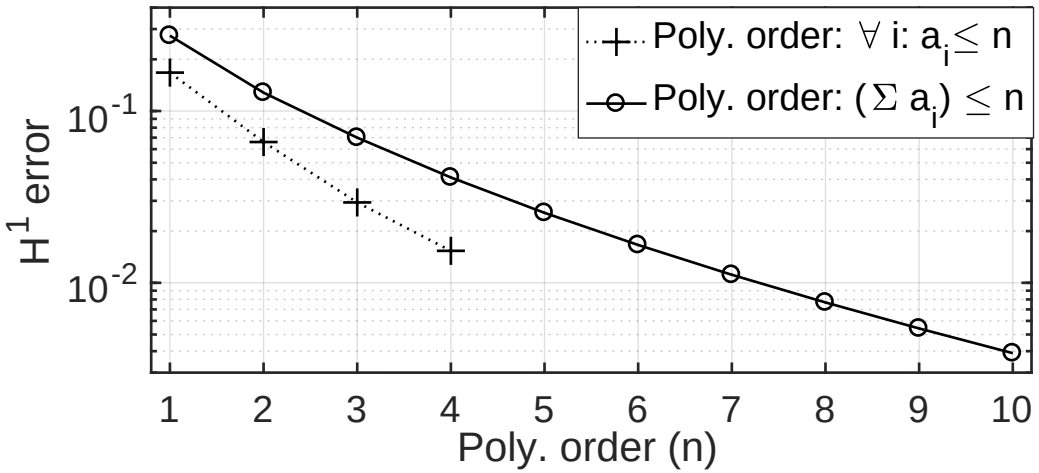


Figure 1: Error dependence on the max. degree of the PC basis

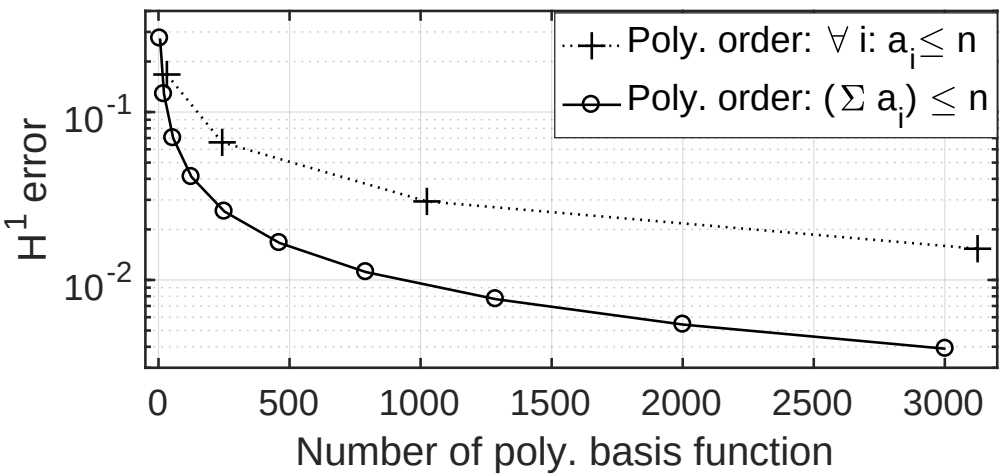


Figure 2: Error dependence on the size of the PC basis

Three term recurrence relation

Let $\{\psi_i\}, \psi_{-1} = 0, \psi_0 = 1$ be polynomials related to probability measure *dFZ* (single random variable), then

$$\psi_{i+1}(Z) = (Z - \alpha_i)\psi_i(Z) - \beta_i \psi_{i-1}(Z),$$

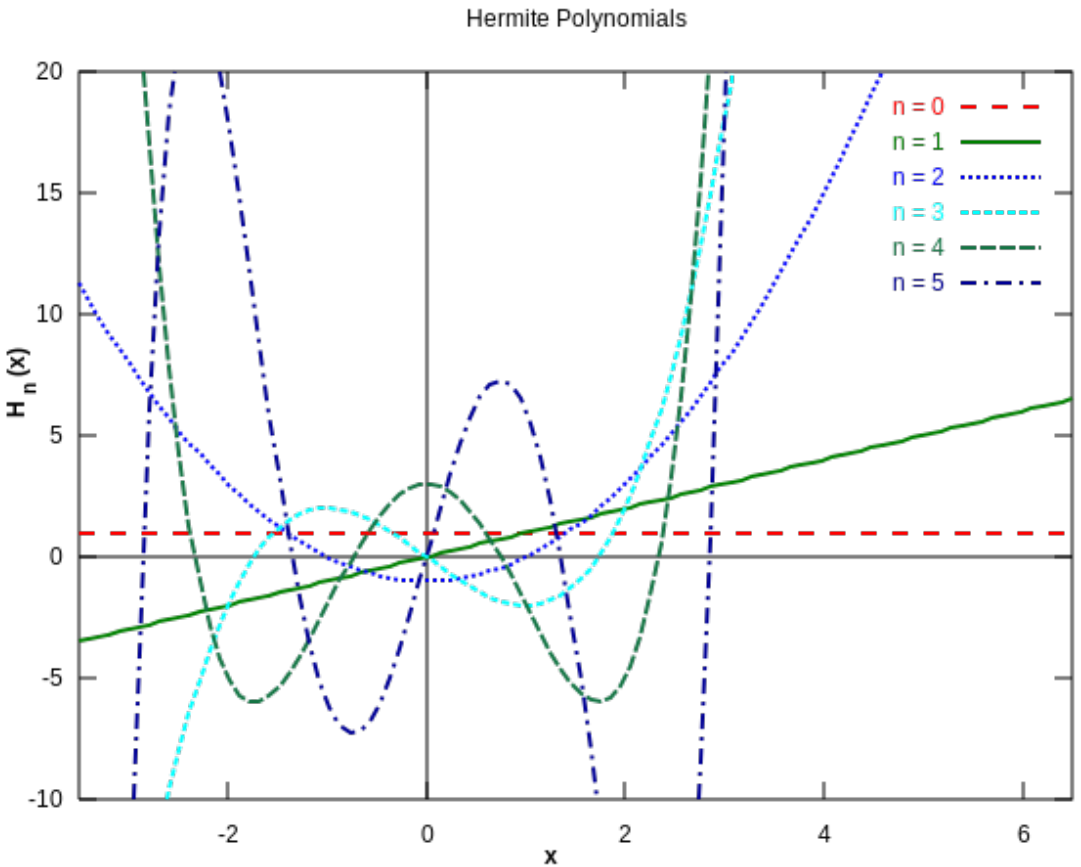
$$\alpha_i = \frac{\mathbb{E}_Z(Z\psi_i(Z)\psi_i(Z))}{\mathbb{E}_Z(\psi_i(Z)\psi_i(Z))}, \beta_i = \frac{\mathbb{E}_Z(\psi_i(Z)\psi_i(Z))}{\mathbb{E}_Z(\psi_{i-1}(Z)\psi_{i-1}(Z))}$$

- evaluation of orthogonal polynomials of higher degree can be numerically unstable
  - do not evaluate weights of the polynomials ($a_0 + a_1 x + \ldots$), numerically unstable (infeasible for degree 20 and more in int32)
  - recurrence formula for computing the point values of the polynomial is favorable

Hermite polynomials

- polynomial chaos basis for standard normal random variable consist of Hermite polynomials
-

$$\psi_{i+1}(Z) = 2Z\psi_i(Z) - i\psi_{i-1}(Z)$$



Structure of the problem, discretization and suitable problems

Model problem again

- 2D Darcy flow problem

$$\begin{cases} -\mathrm{div}\left(k\left(x;\boldsymbol{Z}\right)\cdot\nabla u\left(x;\boldsymbol{Z}\right)\right)=f\left(x;\boldsymbol{Z}\right), & \forall x\in\mathcal{D} \\ u\left(x;\boldsymbol{Z}\right)=u_D\left(x;\boldsymbol{Z}\right), & \forall x\in\Gamma_D\ , \\ \frac{\partial u(x;\boldsymbol{Z})}{\partial n(x)}=u_N\left(x;\boldsymbol{Z}\right), & \forall x\in\Gamma_N \end{cases}$$

$\boldsymbol{Z}$ can be understood as $\boldsymbol{Z}:\Omega\to\mathbb{R}^M$ and $k\left(x;\boldsymbol{Z}\right)$,
$u\left(x;\boldsymbol{Z}\right):\mathcal{D}\times\mathbb{R}^M\to\mathbb{R}$

Coupled and uncoupled systems

- randomness in f and boundary conditions affects the right hand side
  - if k is deterministic, we obtain decoupled system = matrix of the resulting system of equations is block diagonal
- randomness in material k affects the matrix of the resulting system of equations
  - we obtain coupled system

Model problem

- 2D Darcy flow problem

$$\begin{cases} -\mathrm{div}\left(k\left(x;\boldsymbol{Z}\right)\cdot\nabla u\left(x;\boldsymbol{Z}\right)\right)=f\left(x\right), & \forall x\in\mathcal{D} \\ u\left(x;\boldsymbol{Z}\right)=u_D\left(x\right), & \forall x\in\Gamma_D\ , \\ \frac{\partial u(x;\boldsymbol{Z})}{\partial n(x)}=u_N\left(x\right), & \forall x\in\Gamma_N \end{cases}$$

$\boldsymbol{Z}$ can be understood as $\boldsymbol{Z}:\Omega\to\mathbb{R}^M$ and $k\left(x;\boldsymbol{Z}\right)$,
$u\left(x;\boldsymbol{Z}\right):\mathcal{D}\times\mathbb{R}^M\to\mathbb{R}$

Tensor product spaces

- we can view $k(x;\boldsymbol{Z})$, $u(x;\boldsymbol{Z})$ from the perspective of Monte Carlo sampling
  - sample of $\boldsymbol{Z}$ (e.g. $\tilde{Z}$) gives us a sample of $k\left(x;\tilde{Z}\right) \in L^{\infty}(\mathcal{D})$ and $u\left(x;\tilde{Z}\right) \in H^1(\mathcal{D})$
- than $u(x;\boldsymbol{Z}) \in L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M, H^1(\mathcal{D})\right)$

$$L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M, H^1(\mathcal{D})\right) := \left\{ f : \mathbb{R}^M \to H^1(\mathcal{D}) : \int_{\mathbb{R}^M} \|f(\boldsymbol{Z})\|^2_{H^1(\mathcal{D})}\, dF\boldsymbol{Z} < \infty \right\}$$

- $L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M, H^1(\mathcal{D})\right)$ is isometrically isomorphic with tensor product space $H^1(\mathcal{D}) \otimes L^2_{\mathrm{d}F\boldsymbol{Z}}\left(\mathbb{R}^M\right)$

Weak formulation, separable structure

- we want to find the solution in the space $V = V_D \otimes V_S$ $(V_D = H^1(\mathcal{D}), V_S = L^2_{\mathrm{d}F\boldsymbol{Z}}(\mathbb{R}^M))$
- we need to solve the following

$$\begin{cases} \text{Find } u = \sum_{i=1}^{\infty} u_{D,i} \cdot u_{S,i} \in V_D \otimes V_S, \forall v_D \in V_D, v_S \in V_S : \\ a(u,v) = b(v), v = v_D \cdot v_S \end{cases},$$

$$a(u,v) := a(u_D \cdot u_S, v_D \cdot v_S) = \sum_{r=1}^{R} a_{D,r}(u_D, v_D) \cdot a_{S,r}(u_S, v_S)$$

$$b(v) := b(v_D \cdot v_S) = \sum_{r=1}^{S} b_{D,r}(v_D) \cdot b_{S,r}(v_S)$$

Model problem and Galerkin discretization

$$a(u,v) := \int_{\mathbb{R}^M} \int_{\mathcal{D}} k(x;\boldsymbol{Z}) \cdot \nabla_x u(x;\boldsymbol{Z}) \cdot \nabla_x v(x;\boldsymbol{Z}) \,\mathrm{d}x \,\mathrm{d}F\boldsymbol{Z},$$

$$b(v) := \int_{\mathbb{R}^M} \int_{\mathcal{D}} f(x) \cdot v(x;\boldsymbol{Z}) \,\mathrm{d}x \,\mathrm{d}F\boldsymbol{Z},$$

- we use discretization as tensor product of basis functions on domain and basis functions on sample space
  $\langle \varphi_1(x), \ldots, \varphi_{N_D}(x) \rangle \otimes \langle \psi_1(\boldsymbol{Z}), \ldots, \psi_{N_P}(\boldsymbol{Z}) \rangle \subset V_H$:
  $\langle \varphi_1(x), \ldots, \varphi_{N_D}(x) \rangle \subset H_0^1(\mathcal{D})$ and
  $\langle \psi_1(\boldsymbol{Z}), \ldots, \psi_{N_P}(\boldsymbol{Z}) \rangle \subset L^2_{\mathrm{d}F\boldsymbol{Z}}(\mathbb{R}^M)$
- As basis functions on physical domain we can take standard finite elements functions (piece-wise linear)
- For basis functions on sample (random parameter) space we choose polynomial chaos basis

Preferable forms of material field

- recall the

$$a(u,v) := \int_{\mathbb{R}^M} \int_{\mathcal{D}} k(x;\boldsymbol{Z}) \cdot \nabla_x u(x;\boldsymbol{Z}) \cdot \nabla_x v(x;\boldsymbol{Z}) \, \mathrm{d}x \, \mathrm{d}F\boldsymbol{Z}$$

- if we have separable form of the material field

$$k(x;\boldsymbol{Z}) = \sum_{m=1}^{N} k_m(x) \cdot g_m(\boldsymbol{Z})$$

- than resulting system of equation is also separable

$$(\mathbb{A})_{ij,kl} = \sum_{m=1}^{N} \left(\int_{\mathcal{D}} k_m(x) \nabla\varphi_j(x) \nabla\varphi_l(x) \, \mathrm{d}x \right) \cdot \left(\int_{\mathbb{R}^M} g_m(\boldsymbol{Z}) \psi_i(\boldsymbol{Z}) \psi_k(\boldsymbol{Z}) \, \mathrm{d}F\boldsymbol{Z} \right),$$

Preferable forms of material field, how to achieve them

- it is natural for the Karhunen-Loève decomposition of the random field

$$k\left(x;\boldsymbol{Z}\right)=\sum_{m=1}^{M}k_i\left(x\right)\cdot Z_i$$

  or material field with given interfaces

- in other cases it can be obtained by a projection into polynomial chaos

$$k\left(x;\boldsymbol{Z}\right)\approx\sum_{m=1}^{N}k_i\left(x\right)\cdot\Psi_i\left(\boldsymbol{Z}\right)$$

System of equations and assembly

Out-coming system of equations - model problem

- After discretization we get following Galerkin system of equations:

$$\mathbb{A} \cdot \overline{u_H} = \overline{b},$$

$$(\mathbb{A})_{ij,kl} = \sum_{m=1}^{N} \int_{\mathcal{D}} k_m(x) \nabla \varphi_j(x) \nabla \varphi_l(x) \, \mathrm{d}x \cdot \int_{\mathbb{R}^M} g_m(\boldsymbol{Z}) \psi_i(\boldsymbol{Z}) \psi_k(\boldsymbol{Z}) \, \mathrm{d}F\boldsymbol{Z},$$

$$\mathbb{A} = \sum_{m=1}^{N} G_m \otimes K_m,$$

$$(K_m)_{ij} = \int_{\mathcal{D}} k_m(x) \nabla \varphi_i(x) \cdot \nabla \varphi_j(x) \, \mathrm{d}x,$$

$$(G_m)_{ij} = \int_{\mathbb{R}^M} g_m(\boldsymbol{Z}) \psi_i(\boldsymbol{Z}) \psi_j(\boldsymbol{Z}) \, \mathrm{d}F\boldsymbol{Z}.$$

Out-coming system of equations - model problem

$$\overline{b} = \overline{g} \otimes \overline{f}$$

$$(\overline{g})_i = \int_{\mathbb{R}^M} \psi_i(\boldsymbol{Z})\, \mathrm{d}F\boldsymbol{Z}$$

in this case $\overline{g}$ has only one nonzero value in the first position

$$\left(\overline{f}\right)_i = \int_{\mathcal{D}} f(x) \cdot \varphi_i(x)\, \mathrm{d}x$$

- Note: with non-homogeneous boundary conditions the r.h.s will have the general form

$$\overline{b} = \sum_{i=1}^{NN} \overline{g_i} \otimes \overline{f_i}$$

Matrix: stiffness matrix of blocks

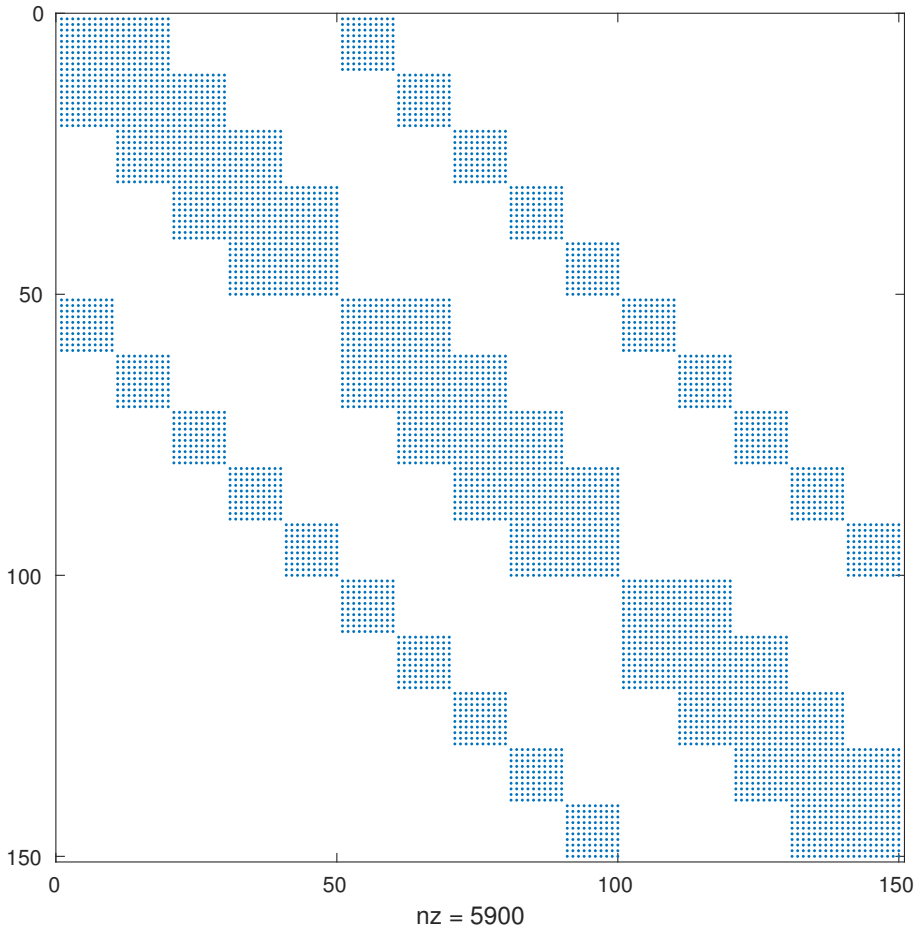


Figure 3: Permuted system matrix for physical domain discretization 4x4 and 10 polynomial chaos basis function

Matrix: blocks of stiffness matrix

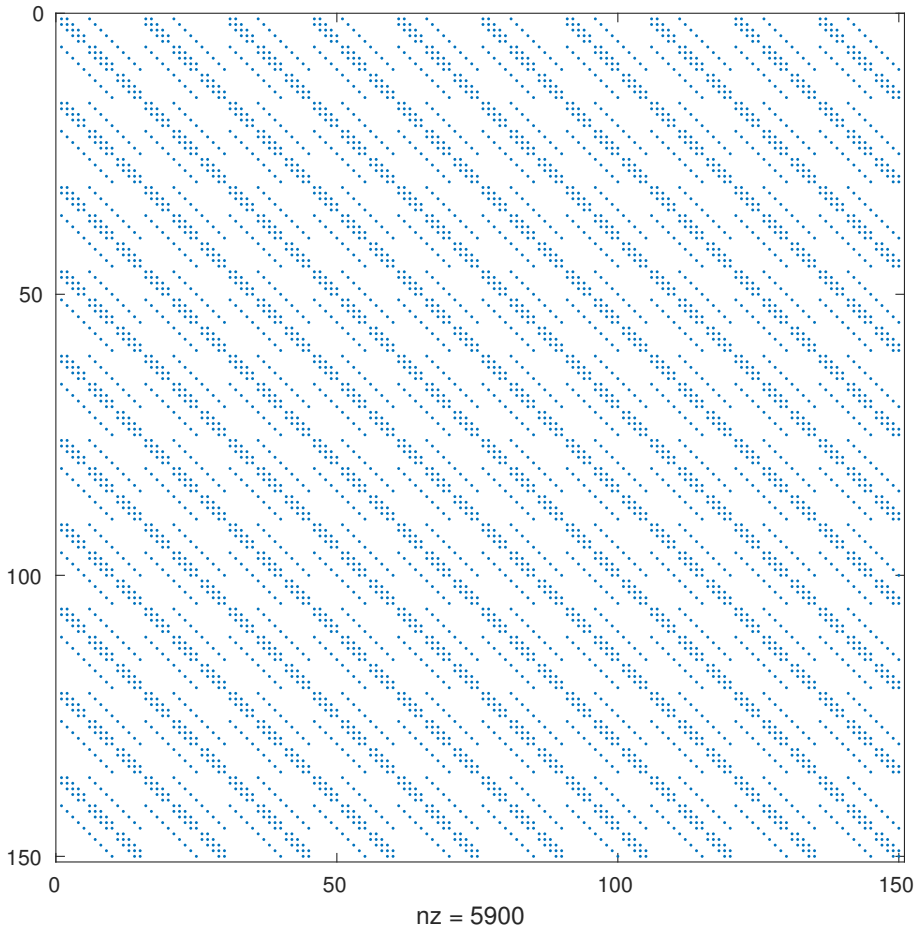


Figure 4: System matrix for physical domain discretization 4x4 and 10 polynomial chaos basis function

Examples of sparsity pattern

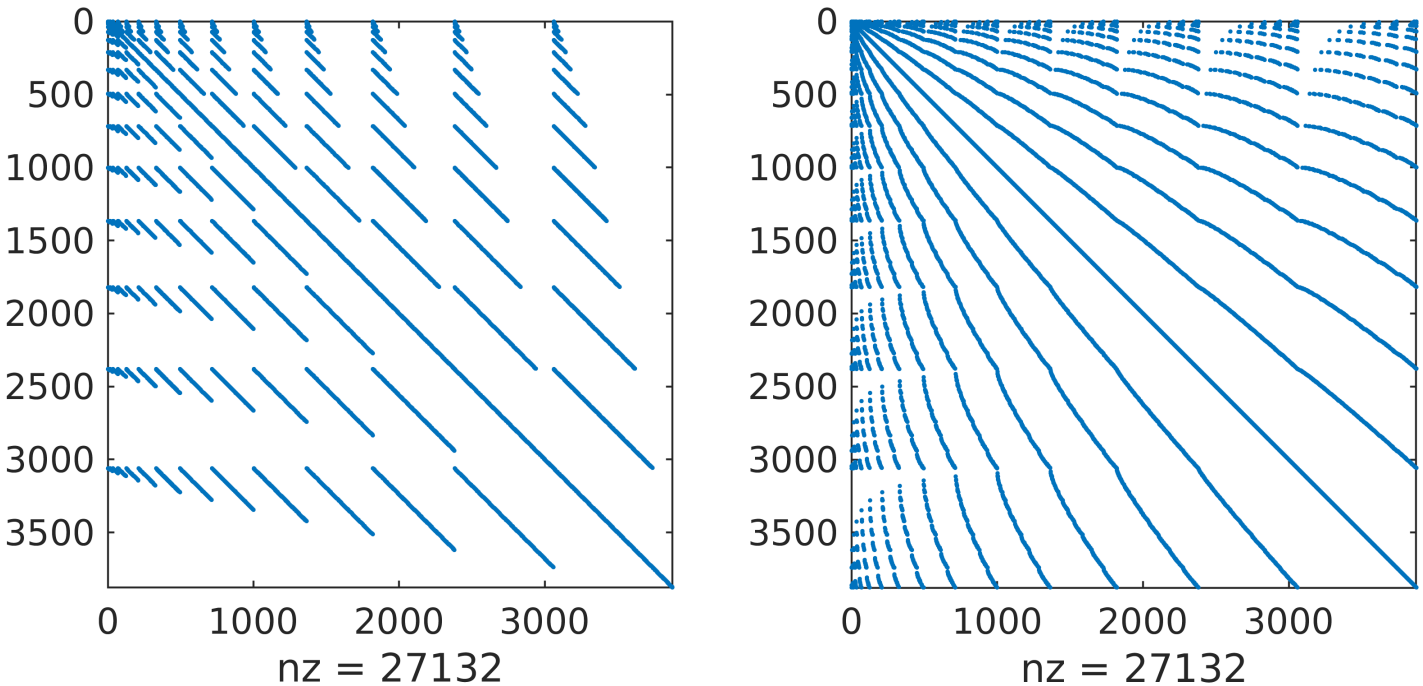


Figure 5: Sparsity pattern of G_i,$g_i = \exp(Z_i)$

Examples of sparsity pattern

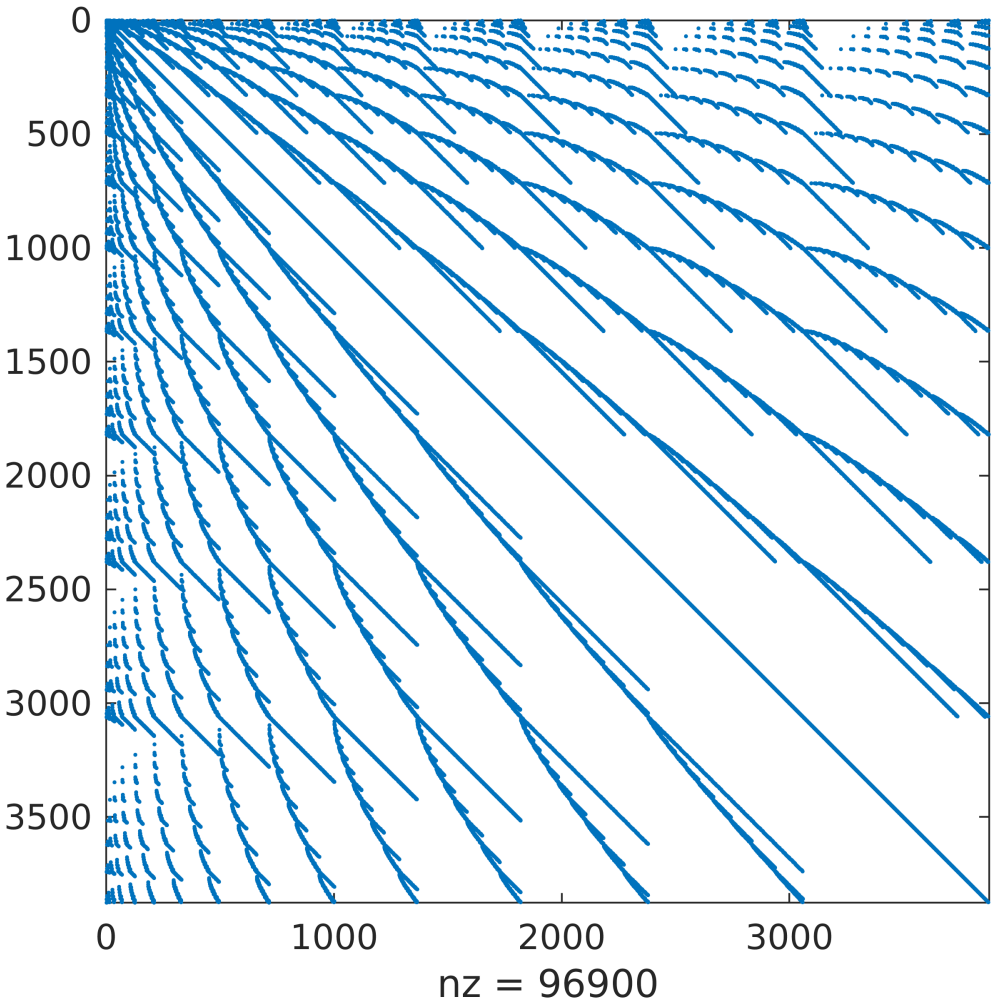


Figure 6: Sparsity pattern of $\sum_i G_i$

Assembly of G_i for $g_i = \exp(Z_i)$

assume normalized Hermite polynomials $\psi_i(\boldsymbol{Z}) = \prod_{j=1}^{M} \psi_{i,j}(Z_j)$

$$(G_m)_{k,l} = \int_{\mathbb{R}^M} \psi_k(\boldsymbol{Z})\, \psi_l(\boldsymbol{Z}) \exp(a_m Z_m + b_m)\, \mathrm{d}F\boldsymbol{Z} =$$

$$\int_{\mathbb{R}} \psi_{k,m}(Z_m)\, \psi_{l,m}(Z_m) \exp(a_m Z_m + b_m)\, \mathrm{d}FZ_m \prod_{i \neq m} \int_{\mathbb{R}} \psi_{k,i}(Z_i)\, \psi_{l,i}(Z_i)\, \mathrm{d}FZ_i,$$

where $\prod_{i \neq m} \int_{\mathbb{R}} \psi_{k,i}(Z_i)\, \psi_{l,i}(Z_i)\, \mathrm{d}FZ_i$ is either 0 if some of the polynomials $\psi_{k,i}$ do not equal $\psi_{l,i}$ or 1 otherwise. The rest of the formula can be evaluated according to the following

$$\int_{\mathbb{R}} f(Z) \exp(aZ + b)\, \mathrm{d}FZ = \int_{\mathbb{R}} f(Z) \exp(aZ + b) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{Z^2}{2}\right) \mathrm{d}Z =$$

$$\exp\left(\frac{a^2}{2} + b\right) \int_{\mathbb{R}} f(Z) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(Z-a)^2}{2}\right) \mathrm{d}Z = \exp\left(\frac{a^2}{2} + b\right) \int_{\mathbb{R}} f(Z)\, \mathrm{d}F\tilde{Z},$$

where $\tilde{Z} \sim \mathcal{N}(a, 1)$.

$\int_{\mathbb{R}} f(Z)\, \mathrm{d}F\tilde{Z}$ can be precisely computed using the Gauss-Hermite quadrature rule or evaluated analytically (if f is polynomial), but the analytic formula can be numerically unstable.

Tensor form matrix multiplication

$$\mathbb{A} = \sum_{m=1}^{N} G_m \otimes K_m,$$

Matrix multiplication

$$\mathbb{A}x = \text{vec}\left(\sum_{m=1}^{N} K_m \tilde{x} G_m^T\right),$$

where $\tilde{x}$ is reshaped x into a matrix of corresponding dimensions.

Approaches to the solution of the system

Overview of approaches

- Krylov method with a suitable preconditioner with a low-rank compression in each iteration [7, 2, 4, 6]
  - preconditioners: block diagonal [15], Kronecker [18] hierarchical Schur [17]
- Generalized Spectral Decomposition, where the problem is treated as an extended eigenvalue problem and the solution is built using the power-type method [10, 9, 14, 11, 13, 12] (even nonlinear problems)
- low rank tensor approximation (also for multiple independent parameters): approximates the solution using the tensor train or hierarchical Tucker format [5, 1, 3].
- reduced basis (RB) approach - construction of the RB using the rational Krylov approximations [16]; using MC sampling [19], using sampling via sparse grids is presented in [8]

Reduced basis

- assume a low rank approximation of the solution

$$\boldsymbol{x} \approx \boldsymbol{x_k} = W_k \boldsymbol{y_k},$$

where $W_k = [w_1, \ldots, w_k] \in \mathbb{R}^{N_d \times k}$ is a **Reduced basis** of size k (k orthonormal vectors)

  - here W_k approximates "deterministic" part of the solution $\boldsymbol{x}$ (given by matrices K_m)

- matrix $\boldsymbol{y_k}$ can be obtained using the Galerkin condition on the residual of $\boldsymbol{x_k}$

$$R_k := \sum_{m=1}^{M} K_m \left(W_k y_k\right) G_m^T - \sum_{m=1}^{M} f_m g_m^T$$

$$W_k^T R_k = 0 \Rightarrow \sum_{m=1}^{M} W_k^T K_m W_k y_k G_m^T = \sum_{m=1}^{M} W_k^T f_m g_m^T$$

which is a system of equations of much lower dimension (can be easily solved by e.g. PCG)

Bibliography i

[1] Jonas Ballani and Lars Grasedyck.
A projection method to solve linear systems in tensor format.
Numerical Linear Algebra with Applications, 20(1):27–43, January 2013.

[2] Peter Benner, Akwum Onwunta, and Martin Stoll.
Low-Rank Solution of Unsteady Diffusion Equations with Stochastic Coefficients.
SIAM/ASA Journal on Uncertainty Quantification, 3(1):622–649, January 2015.

[3] Sergey Dolgov, Boris N. Khoromskij, Alexander Litvinenko, and Hermann G. Matthies.
Polynomial Chaos Expansion of Random Coefficients and the Solution of Stochastic Partial Differential Equations in the Tensor Train Format.

SIAM/ASA Journal on Uncertainty Quantification, 3(1):1109–1135, January 2015.

[4] Howard C. Elman and Tengfei Su.
A Low-Rank Multigrid Method for the Stochastic Steady-State Diffusion Problem.
arXiv:1612.05496 [math], December 2016.
arXiv: 1612.05496.

[5] Boris N. Khoromskij and Christoph Schwab.
Tensor-Structured Galerkin Approximation of Parametric and Stochastic Elliptic PDEs.
SIAM Journal on Scientific Computing, 33(1):364–385, January 2011.

Bibliography iii

[6] Kookjin Lee and Howard C. Elman.
A Preconditioned Low-Rank Projection Method with a Rank-Reduction Scheme for Stochastic Partial Differential Equations.
arXiv:1605.05297 [math], May 2016.
arXiv: 1605.05297.

[7] Hermann G. Matthies and Elmar Zander.
Solving stochastic systems with low-rank tensor compression.
Linear Algebra and its Applications, 436(10):3819–3838, May 2012.

[8] Craig J. Newsum and Catherine E. Powell.
Efficient Reduced Basis Methods for Saddle Point Problems with Applications in Groundwater Flow.
SIAM/ASA Journal on Uncertainty Quantification, 5(1):1248–1278, January 2017.

Bibliography iv

[9] Anthony Nouy.
Construction of generalized spectral bases for the approximate resolution of stochastic problems.
Mecanique et Industries, 8(3):283–288, 2007.

[10] Anthony Nouy.
A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations.
Computer Methods in Applied Mechanics and Engineering, 196(45-48):4521–4537, September 2007.

[11] Anthony Nouy.
Generalized spectral decomposition method for solving stochastic finite element equations: Invariant subspace problem and dedicated algorithms.
Computer Methods in Applied Mechanics and Engineering, 197(51-52):4718–4736, October 2008.

Bibliography v

[12] Anthony Nouy.
Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations.
Archives of Computational Methods in Engineering, 16(3):251–285, 2009.

[13] Anthony Nouy.
Proper Generalized Decompositions and Separated Representations for the Numerical Solution of High Dimensional Stochastic Problems.
Archives of Computational Methods in Engineering, 17(4):403–434, December 2010.

[14] Anthony Nouy and Olivier P. Le Maitre.
Generalized spectral decomposition for stochastic nonlinear problems.
Journal of Computational Physics, 228(1):202–235, January 2009.

[15] C. E. Powell and H. C. Elman.
Block-diagonal preconditioning for spectral stochastic finite-element systems.
IMA Journal of Numerical Analysis, 29(2):350–375, April 2008.

[16] C. E. Powell, D. Silvester, and V. Simoncini.
An Efficient Reduced Basis Solver for Stochastic Galerkin Matrix Equations.
SIAM Journal on Scientific Computing, 39(1):A141–A163, 2017.

[17] Bedřich Sousedík, Roger G. Ghanem, and Eric T. Phipps.
Hierarchical Schur complement preconditioner for the stochastic Galerkin finite element methods.
Numerical Linear Algebra with Applications, 21(1):136–151, January 2014.
arXiv: 1205.1864.

[18] Elisabeth Ullmann.
A Kronecker Product Preconditioner for Stochastic Galerkin Finite Element Discretizations.
SIAM Journal on Scientific Computing, 32(2):923–946, January 2010.

[19] Sebastian Ullmann and Jens Lang.
Stochastic Galerkin reduced basis methods for parametrized linear elliptic PDEs.
arXiv:1812.08519 [math], December 2018.
arXiv: 1812.08519.

Reduced basis solver

Low rank approximation of the solution

- assume a low rank approximation of the solution

$$\boldsymbol{x} \approx \boldsymbol{x_k} = W_k \boldsymbol{y_k},$$

where $W_k = [w_1, \ldots, w_k] \in \mathbb{R}^{N_d \times k}$ is a **Reduced basis** of size k (k orthonormal vectors)

  - here W_k approximates "deterministic" part of the solution $\boldsymbol{x}$ (given by matrices K_m)

- matrix $\boldsymbol{y_k}$ can be obtained using the Galerkin condition on the residual of $\boldsymbol{x_k}$

$$R_k := \sum_{m=1}^{M} K_m \left(W_k y_k\right) G_m^T - \sum_{m=1}^{M} f_m g_m^T$$

$$W_k^T R_k = 0 \Rightarrow \sum_{m=1}^{M} W_k^T K_m W_k y_k G_m^T = \sum_{m=1}^{M} W_k^T f_m g_m^T$$

Reduced rational Krylov subspaces

Reduced rational Krylov subspace (RRKS)

- series of symmetric positive definite (SPD) matrices $\{K_m\}_{m=1,\dots,M}$ and a nonzero vector v
- the first iteration generates addition to the basis:

$$\left\langle K_1^{-1} v, \dots, K_M^{-1} v \right\rangle$$

- the second iteration generate addition to the basis:

$$\left\langle K_1^{-1} K_1^{-1} v, K_1^{-1} K_2^{-1} v, \dots, \; K_M^{-1} K_{M-1}^{-1} v, K_M^{-1} K_M^{-1} v \right\rangle$$

Building the reduced basis

- starting vector from $\hat{f}_m = K_0^{-1} f_m$
- using matrices $\left\{K_0^{-1}\left(K_m + \alpha_m K_0\right)\right\}$
- full RRKS scheme is impractical because in each subsequent iteration we need to construct larger bases
- the remedy to this is to iteratively select a vector v from the current basis and expand the basis only by the first iteration of the RRKS
- during the orthogonalisation step, calculate weights corresponding to the norm of independent part of the vector
  - next direction according to the calculated weights


Powell, C. E.; Silvester, D.; Simoncini, V.: An efficient reduced basis solver for stochastic Galerkin matrix equations. 2017


Béreš, M.: An Efficient Reduced Basis Construction for Stochastic Galerkin Matrix Equations Using Deflated Conjugate Gradients. 2018

RRKS convergence - different α

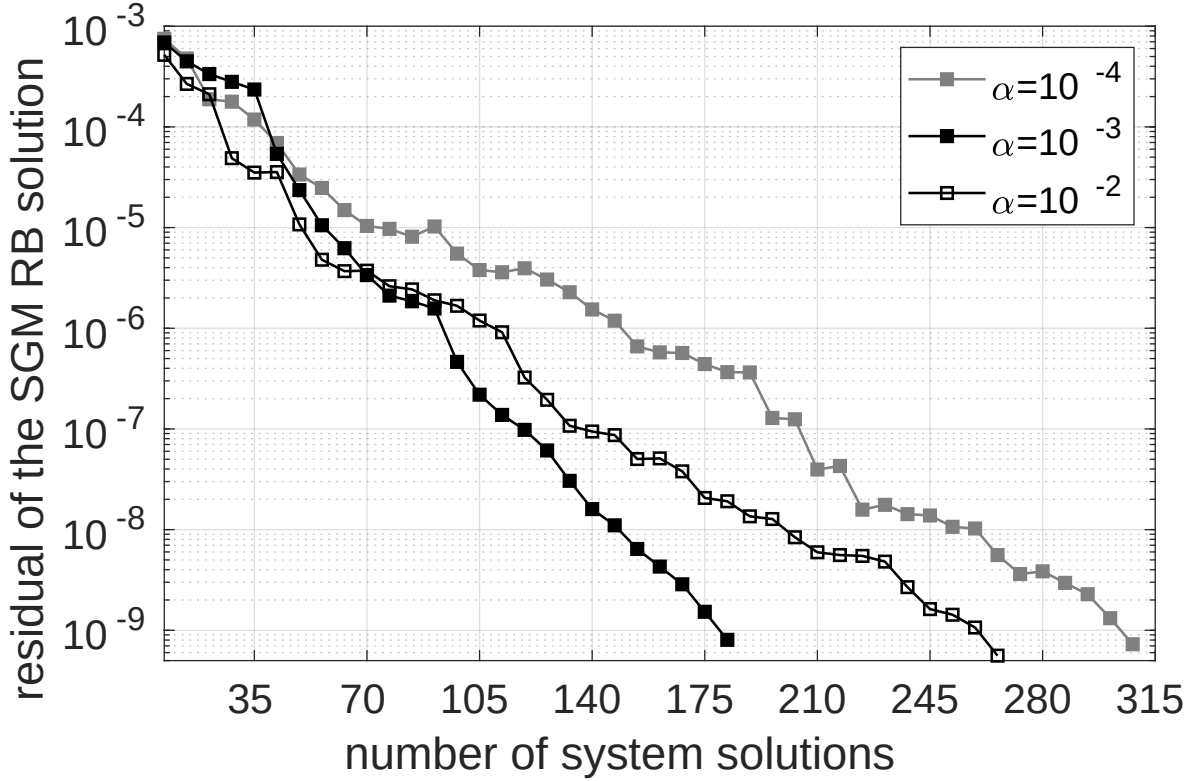


Figure 1: different values of α (degree 4, grid level 10)

RB convergence - polynomial degree

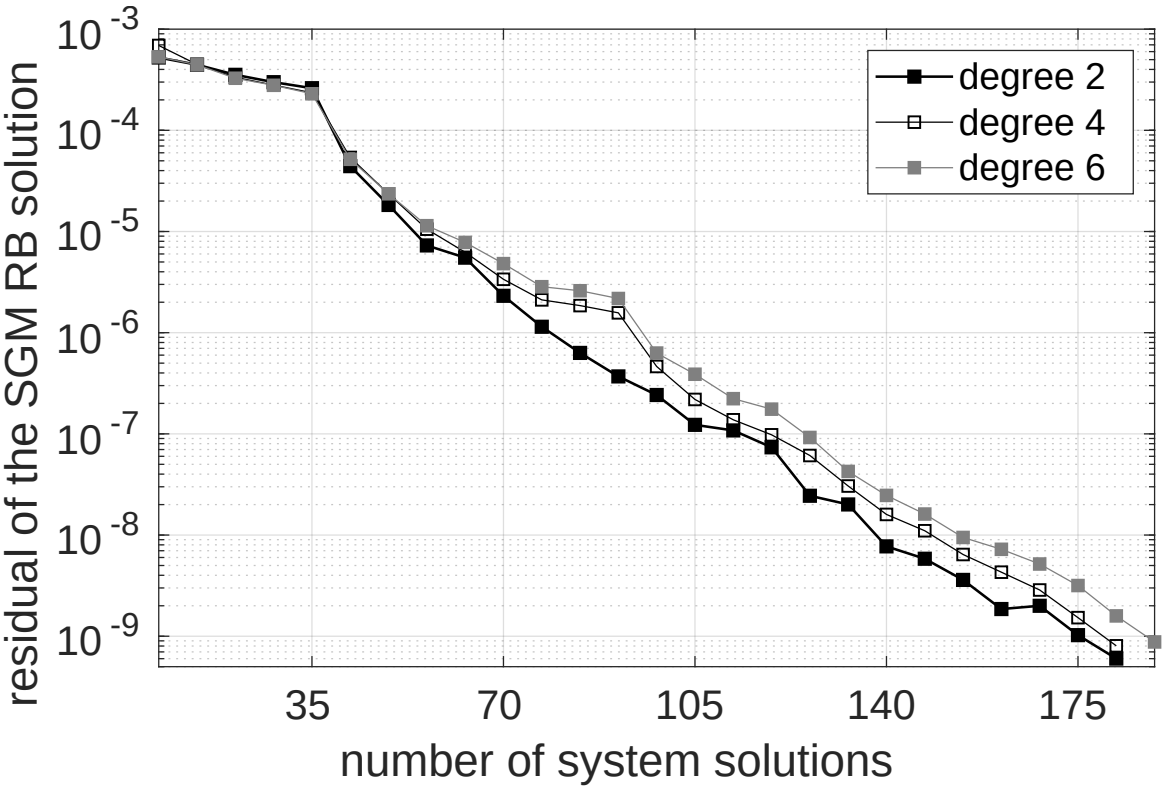


Figure 2: different maximal degree of polynomial chaos (grid lvl 10)

RB convergence - finite element grid

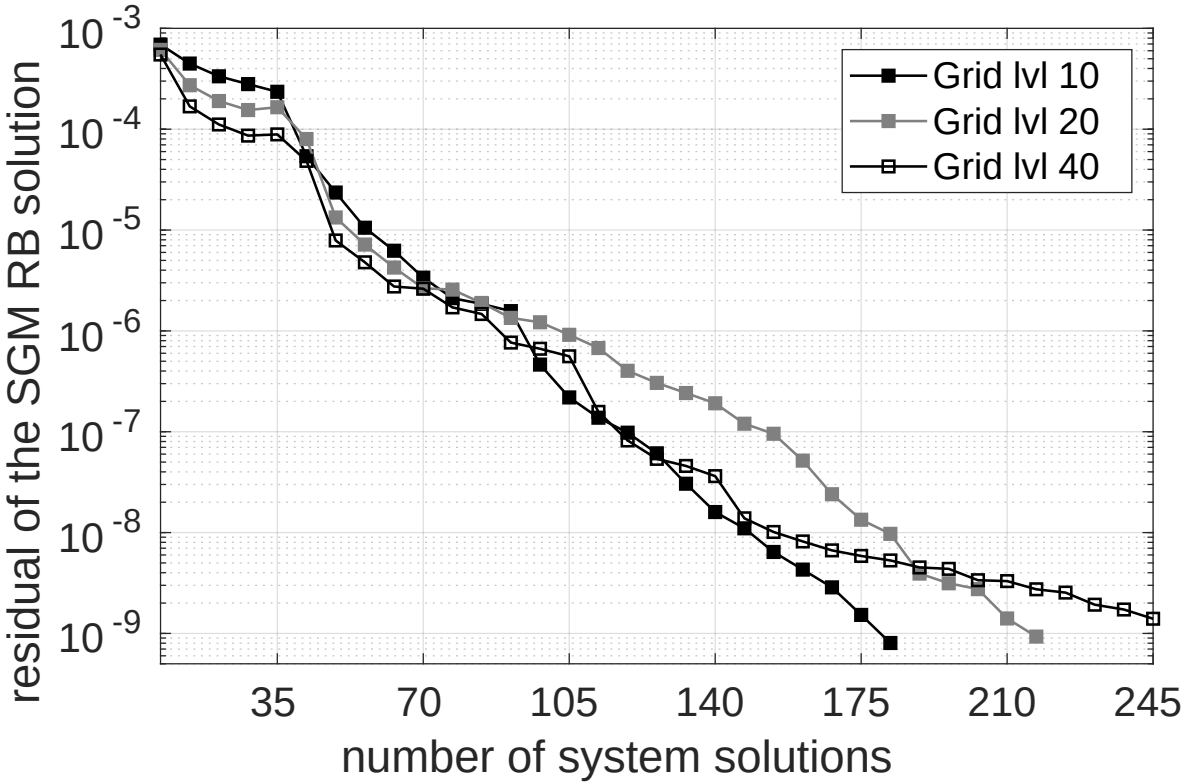


Figure 3: different grid levels (polynomial degree 4)

Monte Carlo sampling

Crude MC and random process avoiding previous samples

We compare two random sampling procedures

- Crude MC
- Sampling using random process, which avoids already generated samples (MCMC)
  - assume sampling according to unscaled probability density $f(x)$
  - each consecutive sample X_i is generated according to unscaled probability density

$$\bar{f}(x) = f(x) \cdot \min \left\{ 1 - \exp\left(-\frac{1}{2} (x - X_j)^T \cdot \Sigma^{-1} \cdot (x - X_j) \right) \right\}_{j=1 \ldots i-1}$$

  - these samples are generated using Metropolis Hastings method (we can take random sample of the chain, or the one with the highest value of $\bar{f}(x)$)

Unscaled probability density after 10 samples

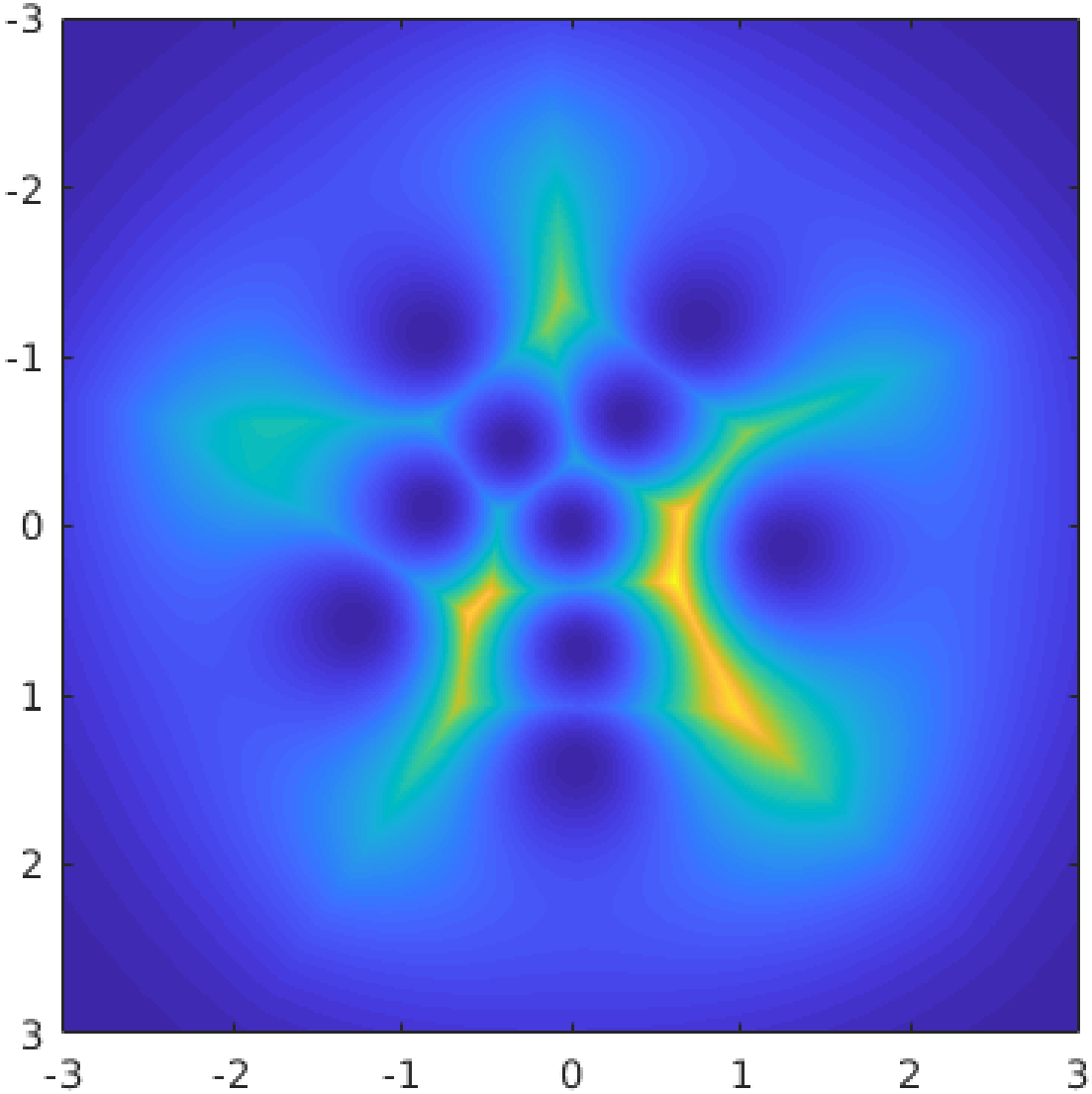


Figure 4: Unscaled probability density after 10 samples ($\Sigma = 10 \cdot I$)

Sample comparison of crude MC and MCMC

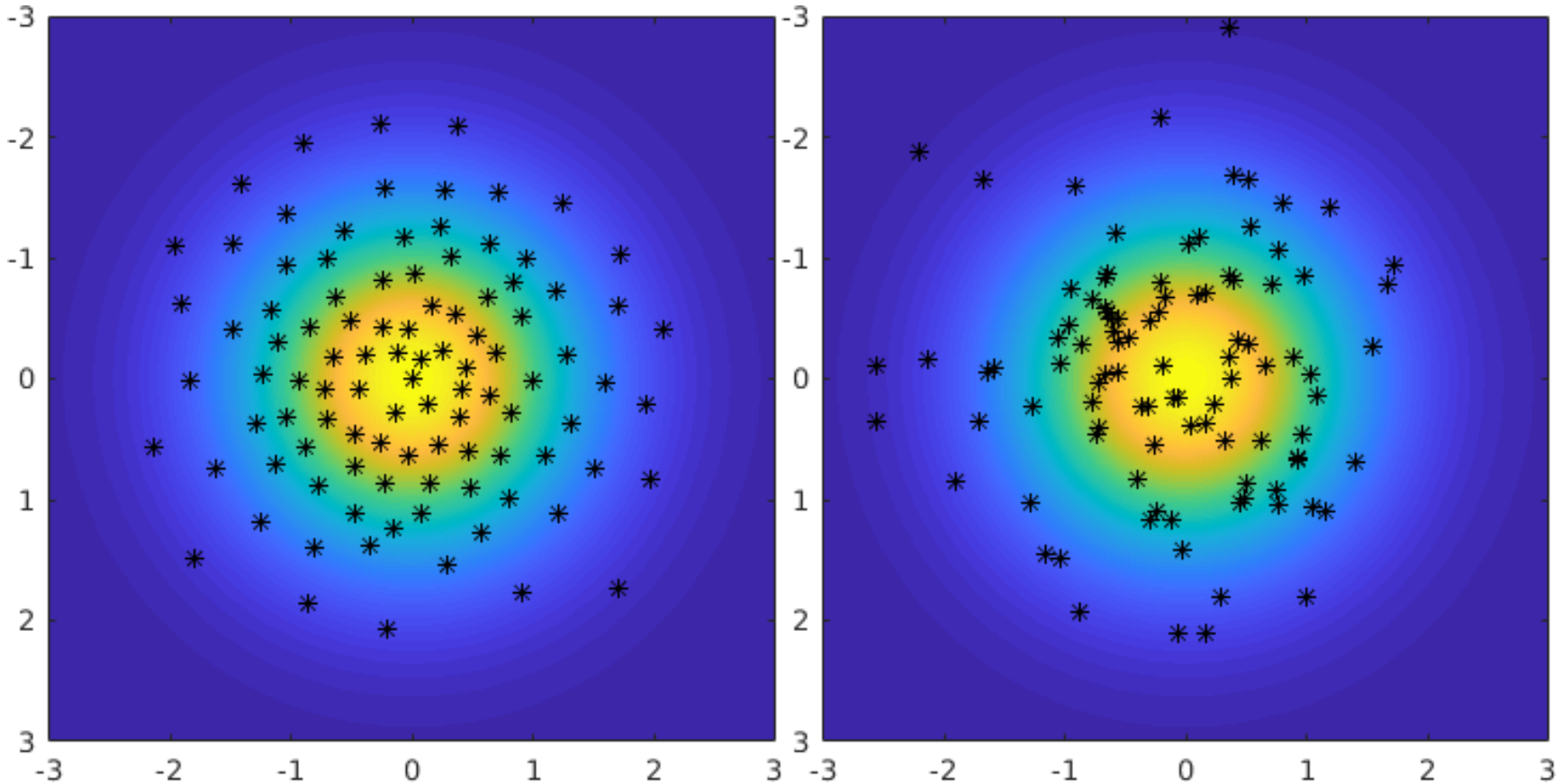


Figure 5: left MCMC, right MC

Reduced basis construction using random sampling

Loop until satisfactory reduced basis is obtained:

1. Propose n samples (MC or MCMC)
2. Calculate residual of reduced solution in this samples
3. Solve systems, using best N samples (largest residual error)
4. Add these N samples into the reduced basis (truncate if needed)

MC vs. MCMC number of proposal samples n

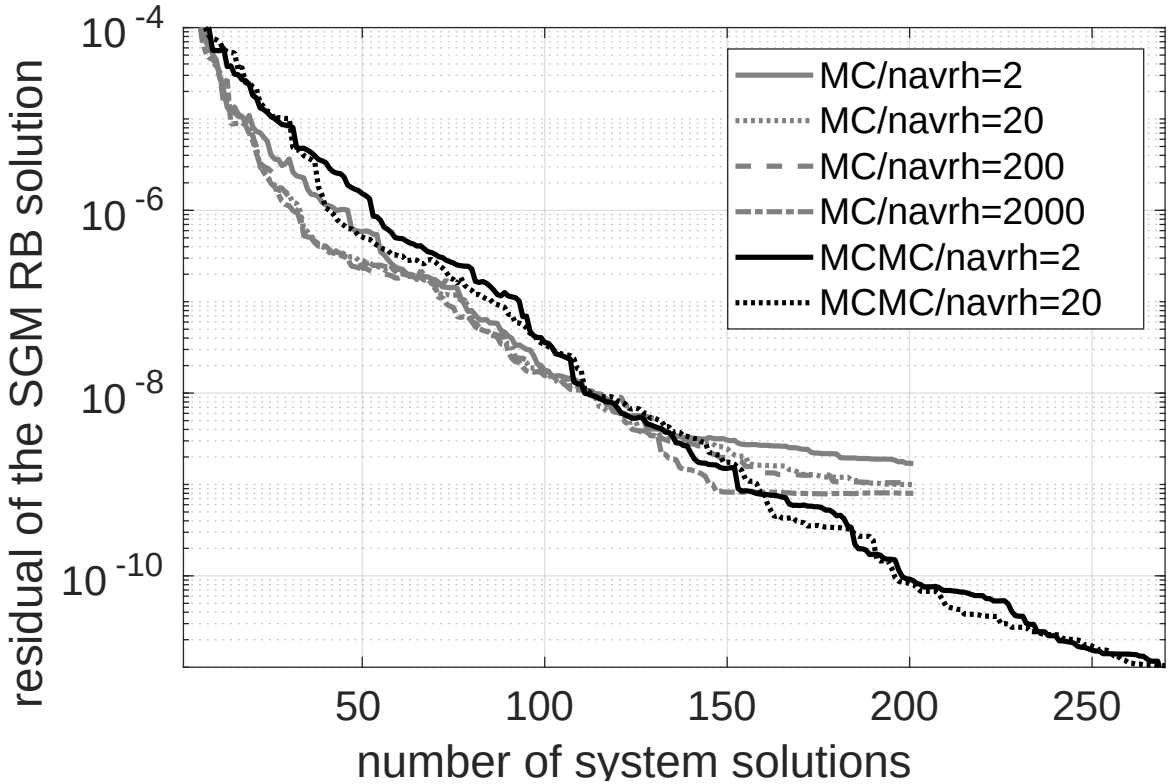


Figure 6: different number of proposal samples n

MC vs. MCMC number of parallel solutions

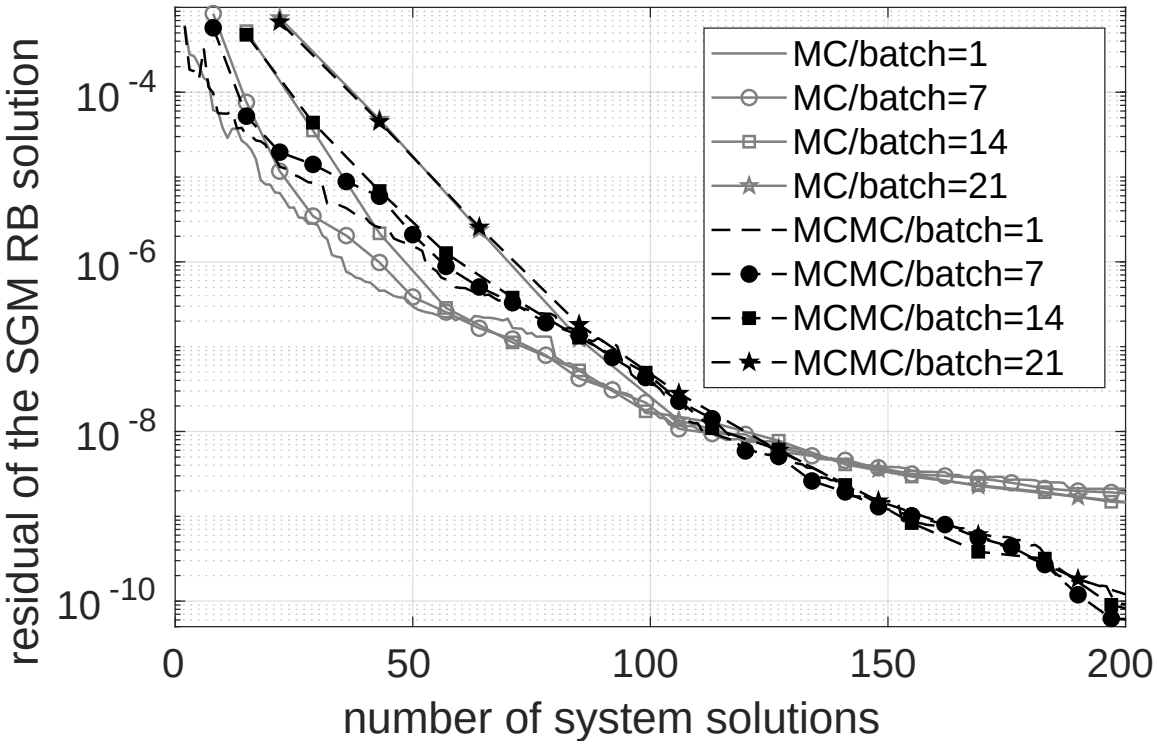


Figure 7: different number of parallel solutions N

MCMC - polynomial degree

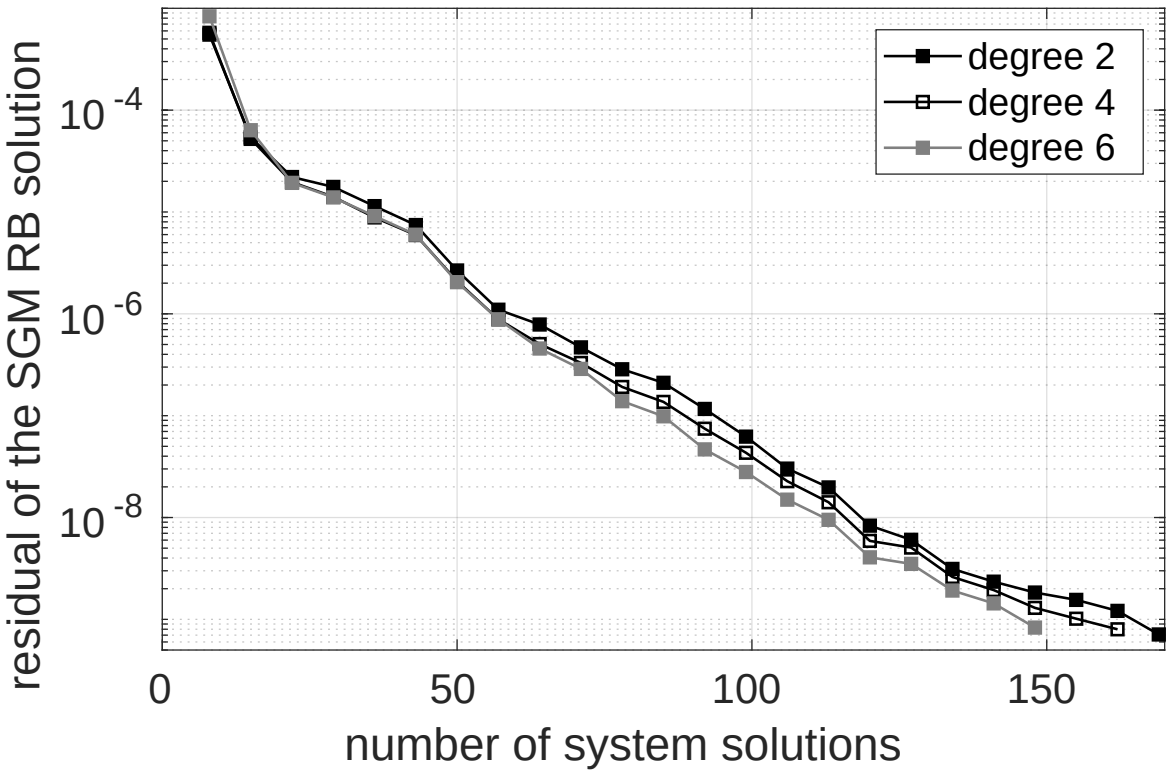


Figure 8: different maximal degree of polynomial chaos (grid lvl 10)

MCMC - grid lvl

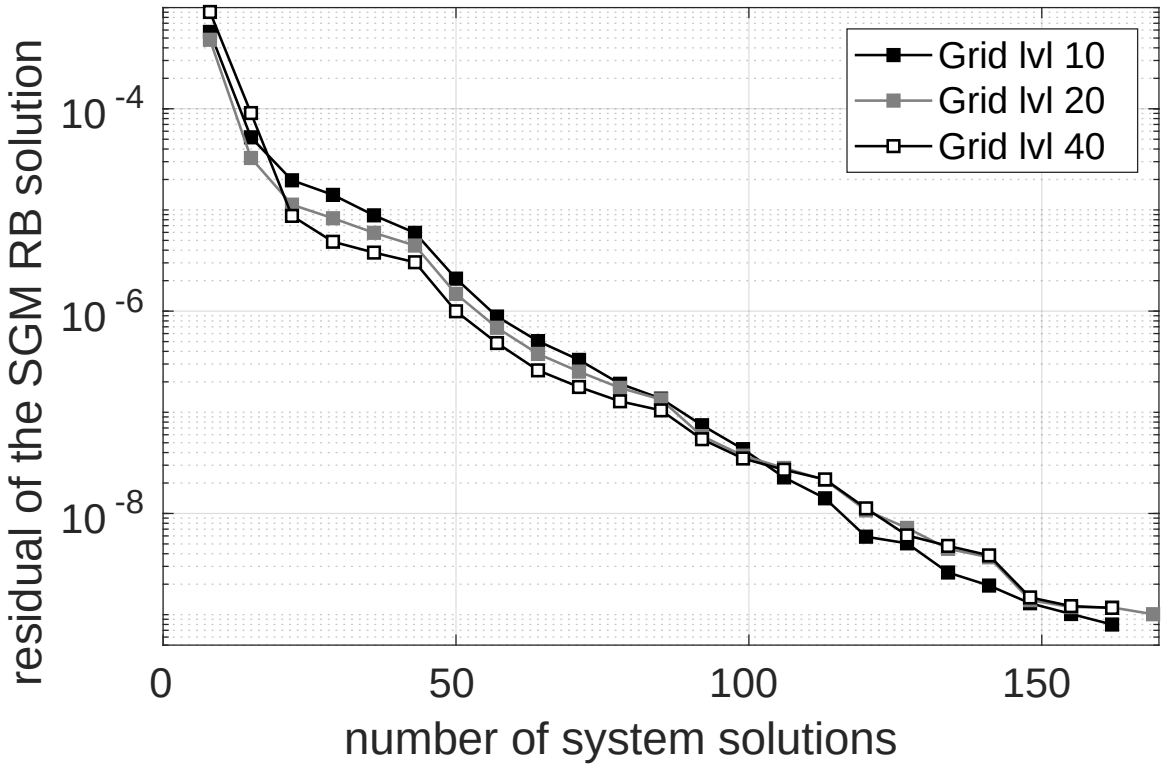


Figure 9: different grid levels (polynomial degree 4)

MCMC vs. RRKS

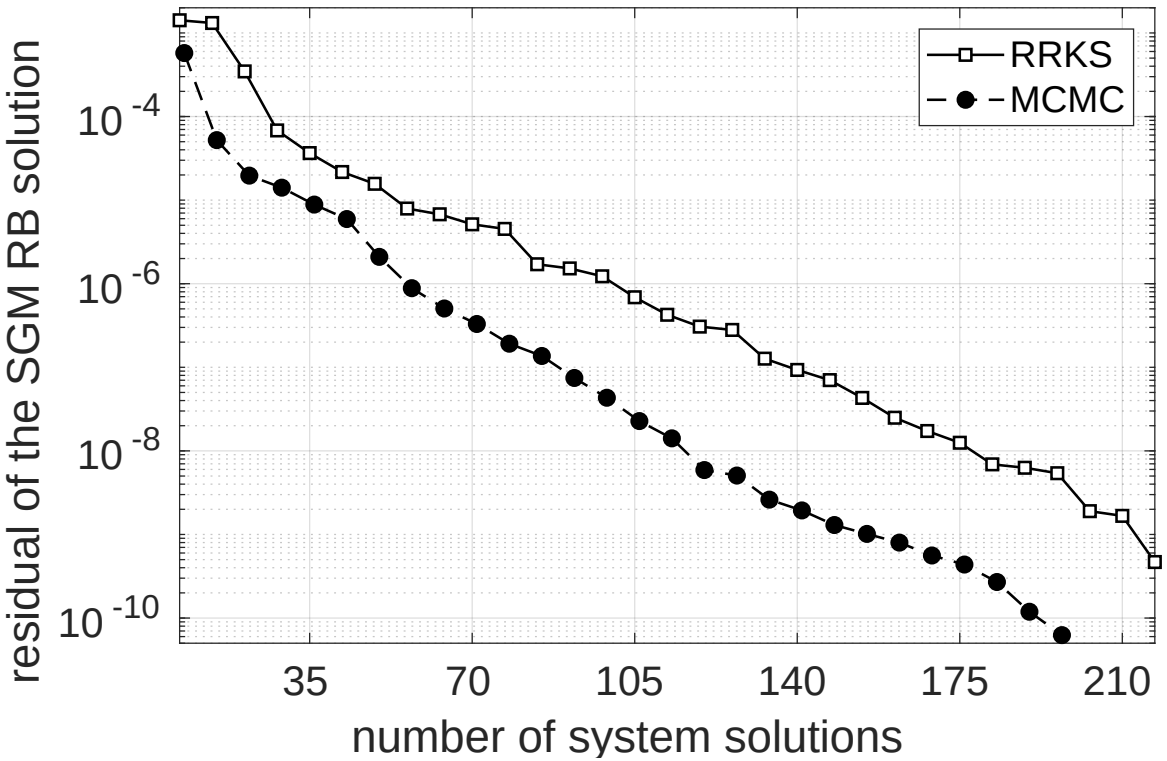


Figure 10: grid lvl 10, polynomial chaos degree 4

Conclusions: MCMC vs. RRKS

MCMC:

- better convergence (same rate, but shifted)
- easier to implement (it is possible to call black box solvers)
- random (different runs can differ)

RRKS:

- it solves the same systems (much cheaper assembly of systems and preconditioners)
- depends on the choice of α (can be difficult)

Deflated conjugate gradients

Deflated conjugate gradients

- DCG method is an extension of the standard conjugate gradient (CG or PCG if using a preconditioner) method
- DCG method takes an additional parameter in the form of the deflation basis W (linearly independent, we will use orthonormal).
- the deflation basis W should be able to describe the sought solution reasonably well
- the DCG method looks for the solution outside of the deflation basis W

Deflated conjugate gradients

- DCG diference in comparison to the conjugate gradient implementation:
  - the residual of the initial guess should be orthogonal on W (e.g. by using $x_0 = W \left(W^T A W\right)^{-1} W^T b$)
  - in each iteration the DCG projects the preconditioned residual $\tilde{z}_i = P z_i$ using the projector $P = I - W \left(W^T A W\right)^{-1} W^T A$
- in our case the reduced basis W_i in i-th iteration of RB solver can be directly used as the deflation basis
- only significant additional cost compared to the standard CG method is the solution of systems with $Q_i = W_i^T A W_i$ (in i-th iteration of the RB solver)
  - In our application, the size of the matrix Q_i (corresponding to a size of the RB) is reasonably small and we can use e.q. an explicit inversion.

Overview of tested preconditioners

- we test the solution of systems $(K_m + \alpha K_0)$ using: Schwarz, diagonal and ichol preconditioner
  - the Schwarz preconditioner was set using 30 subdomains = equal column slices of our square domain of the size $1/20$
  - the incomplete Cholesky preconditioner was build with no filling allowed

RRKS: DCG - results (number of iterations CG/DCG)

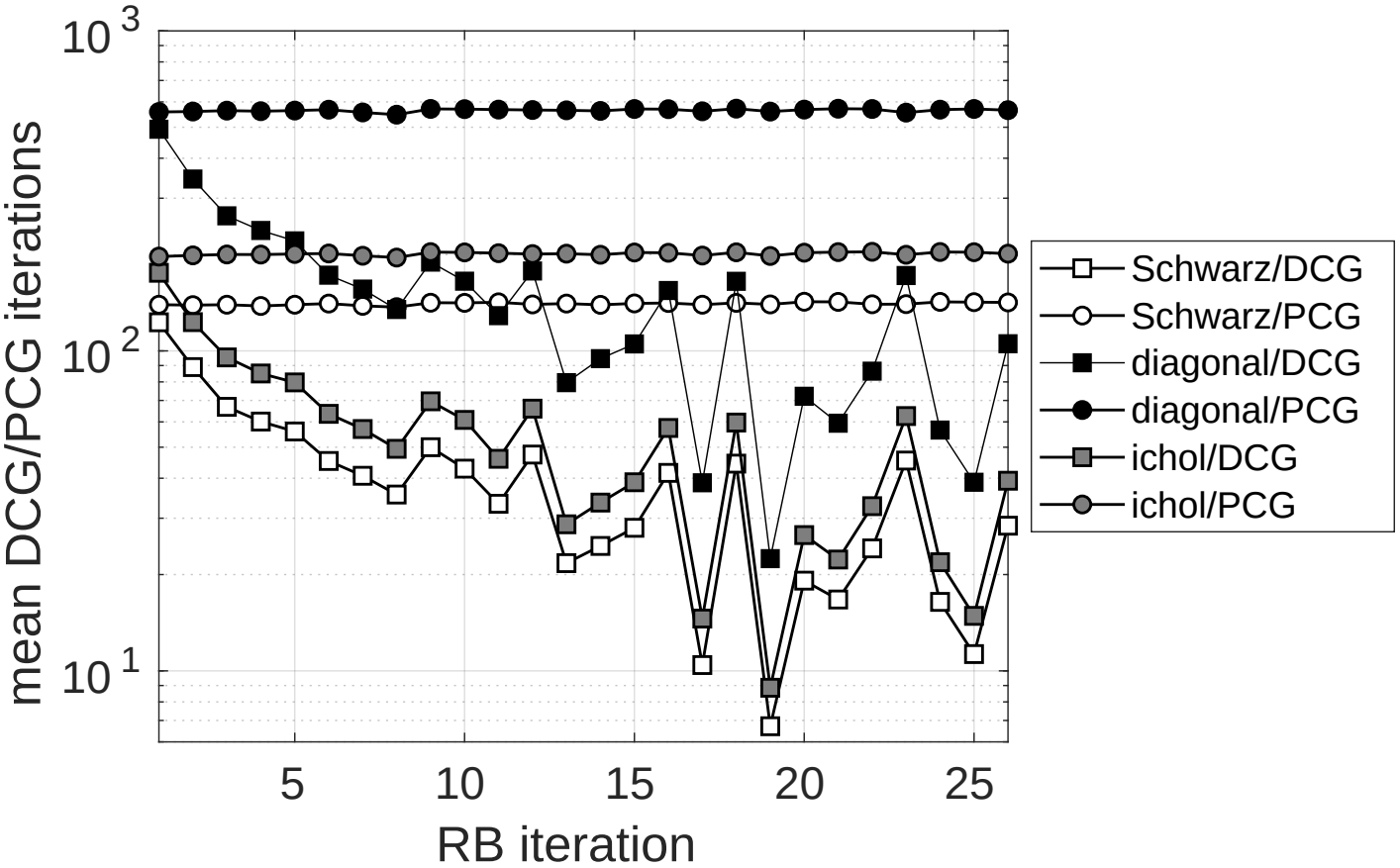


Figure 11: Num. of the DCG iterations compared to the PCG (mean DCG/PCG iterations per RB iter.)

MCMC: DCG - results (number of iterations CG/DCG)

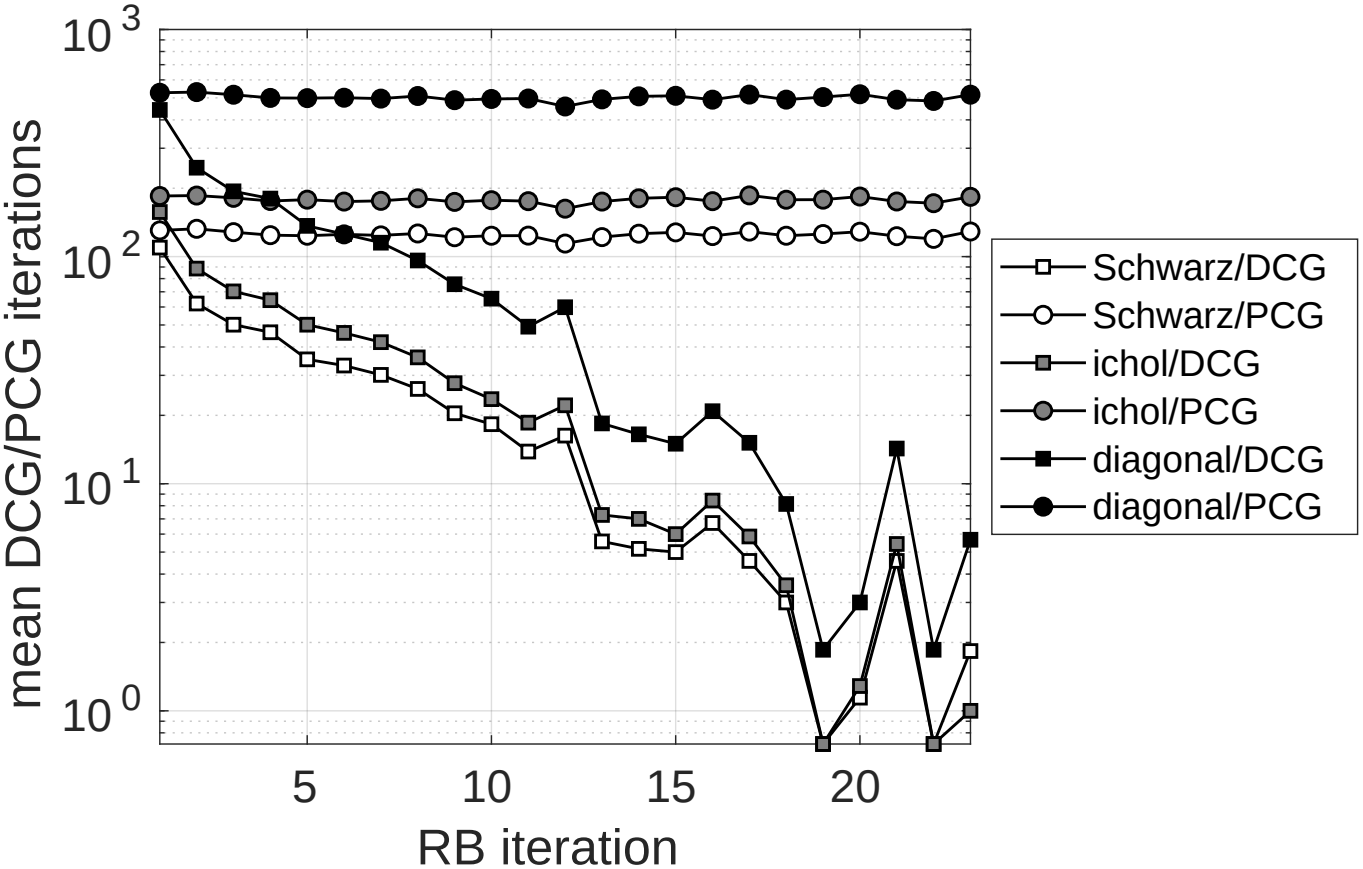


Figure 12: Num. of the DCG iterations compared to the PCG (mean DCG/PCG iterations per RB iter.)

DCG - results (saved iterations in total)

	Ad. Schwarz	**diagonal**	**ichol (nofill)**
RRKS savings in %	72.32%	73.47%	73.33%
MCMC savings in %	82.58%	83.48%	83.06%

Table 1: Computational savings using DCG with the RB as a deflation basis