

# The uniform sparse FFT

with application to PDEs with random coefficients

Fabian Taubert

joint work with Lutz Kämmerer and Daniel Potts

15.09.2021



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

## 1. Introduction

Setting

The sparse FFT

## 2. The uniform sparse FFT

The key idea

Main result

## 3. Numerical examples

Affine random coefficient

Lognormal random coefficient

We consider the PDE problem

$$\begin{aligned}
 -\nabla_{\mathbf{x}} \cdot (a(\mathbf{x}, \mathbf{y}) \nabla_{\mathbf{x}} u(\mathbf{x}, \mathbf{y})) &= f(\mathbf{x}), & \mathbf{x} \in D, \mathbf{y} \in D_{\mathbf{y}} \\
 u(\mathbf{x}, \mathbf{y}) &= 0, & \forall \mathbf{x} \in \partial D, \mathbf{y} \in D_{\mathbf{y}}.
 \end{aligned}$$

- ▶ spatial variable  $\mathbf{x} \in D \subset \mathbb{R}^d$ , typically with  $d = 1, 2, 3$
- ▶ random variable  $\mathbf{y} = (y_j)_{j=1}^{d_{\mathbf{y}}} \in D_{\mathbf{y}}$ , typically very high-dimensional or even infinite-dimensional

- ▶ **affine random coefficient** [Cohen, DeVore, Schwab '10], [Dick, Kuo, Le Gia, Schwab '16], [Bachmayr, Cohen, Dahmen '18], [Gantner, Herrmann, Schwab '18], [Nguyen, Nuyens '21], ...

$$a(\mathbf{x}, \mathbf{y}) = a_0(\mathbf{x}) + \sum_{j=1}^{d_{\mathbf{y}}} y_j \psi_j(\mathbf{x}).$$

- ▶ **periodic random coefficient**

[Kaarnioja, Kuo, Sloan '20], [Kaarnioja, Kazashi, Kuo, Nobile, Sloan '20]

$$a(\mathbf{x}, \mathbf{y}) = a_0(\mathbf{x}) + \sum_{j=1}^{d_{\mathbf{y}}} \Theta_j(\mathbf{y}) \psi_j(\mathbf{x}),$$

with  $\Theta_j$  periodic, e.g.,  $\Theta_j(\mathbf{y}) := \frac{1}{\sqrt{6}} \sin(2\pi y_j)$

- ▶ **lognormal random coefficient** [Graham, Kuo, Nichols, Scheichl, Schwab, Sloan '13], [Cheng, Hou, Yan, Zhang '13], [Bachmayr, Cohen, DeVore, Migliorati '17], [Nguyen, Nuyens '21], ...

$$a(\mathbf{x}, \mathbf{y}) = a_0(\mathbf{x}) + \exp(b(\mathbf{x}, \mathbf{y})), \quad b(\mathbf{x}, \mathbf{y}) = b_0(\mathbf{x}) + \sum_{j=1}^{d_{\mathbf{y}}} y_j \psi_j(\mathbf{x}).$$

- ▶ Common aim: Approximation of the solution  $u(\mathbf{x}, \mathbf{y})$  or some quantity of interest, e.g., the expectation value  $\mathbb{E}[F(\mathbf{y})]$  of some functional  $F(\mathbf{y}) := F[u(\cdot, \mathbf{y})]$
- ▶ Main problem: high-dimensional approximation!

- ▶ Common aim: Approximation of the solution  $u(\mathbf{x}, \mathbf{y})$  or some quantity of interest, e.g., the expectation value  $\mathbb{E}[F(\mathbf{y})]$  of some functional  $F(\mathbf{y}) := F[u(\cdot, \mathbf{y})]$
- ▶ Main problem: high-dimensional approximation!
- ▶ Examples of other approaches:
  - ▶ Quasi-Monte Carlo methods  
[Kuo, Schwab, Sloan '15], [Dick, Le Gia, Schwab '16], [Nguyen, Nuyens '21], ...
  - ▶ collocation methods  
[Cheng, Hou, Yan, Zhang '13], [Ernst, Sprungk '14], [Zhang, Hu, Hou, Lin, Yan '14], ...
  - ▶ methods based on certain (tensorized) functions (e.g., Legendre polynomials) [Cohen, DeVore, Schwab '10], [Bachmayr, Cohen, Migliorati '17], ...

- ▶ Common aim: Approximation of the solution  $u(\mathbf{x}, \mathbf{y})$  or some quantity of interest, e.g., the expectation value  $\mathbb{E}[F(\mathbf{y})]$  of some functional  $F(\mathbf{y}) := F[u(\cdot, \mathbf{y})]$
- ▶ Main problem: high-dimensional approximation!
- ▶ Examples of other approaches:
  - ▶ Quasi-Monte Carlo methods  
[Kuo, Schwab, Sloan '15], [Dick, Le Gia, Schwab '16], [Nguyen, Nuyens '21], ...
  - ▶ collocation methods  
[Cheng, Hou, Yan, Zhang '13], [Ernst, Sprungk '14], [Zhang, Hu, Hou, Lin, Yan '14], ...
  - ▶ methods based on certain (tensorized) functions (e.g., Legendre polynomials) [Cohen, DeVore, Schwab '10], [Bachmayr, Cohen, Migliorati '17], ...
- ▶ These approaches are often heavily influenced by the choice (or computation) of some weights, functions or kernels in advance!

- ▶ We aim for an approximation of  $u(\mathbf{x}_0, \cdot)$  for fixed  $\mathbf{x}_0 \in D$  using the sparse FFT (sFFT) based on rank-1 lattice sampling.

[Potts, Volkmer '16], [Kämmerer, Kraemer, Volkmer '20], [Kämmerer, Potts, Volkmer '21]



- ▶ We aim for an approximation of  $u(\mathbf{x}_0, \cdot)$  for fixed  $\mathbf{x}_0 \in D$  using the sparse FFT (sFFT) based on rank-1 lattice sampling.

[Potts, Volkmer '16], [Kämmerer, Kraher, Volkmer '20], [Kämmerer, Potts, Volkmer '21]

- ▶ The goal of the sFFT is the computation of the index set  $I = \text{supp } \hat{p} \subset \mathbb{Z}^d$  and the coefficients  $\hat{p}_{\mathbf{k}}, \mathbf{k} \in I$ , of the multivariate trigonometric polynomial

$$p(\mathbf{y}) = \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}} e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

from sampling values.

- ▶ We aim for an approximation of  $u(\mathbf{x}_0, \cdot)$  for fixed  $\mathbf{x}_0 \in D$  using the sparse FFT (sFFT) based on rank-1 lattice sampling.

[Potts, Volkmer '16], [Kämmerer, Kraemer, Volkmer '20], [Kämmerer, Potts, Volkmer '21]

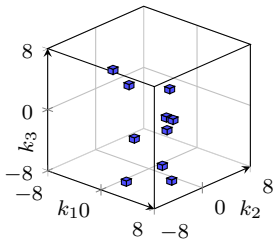
- ▶ The goal of the sFFT is the computation of the index set  $I = \text{supp } \hat{p} \subset \mathbb{Z}^d$  and the coefficients  $\hat{p}_{\mathbf{k}}, \mathbf{k} \in I$ , of the multivariate trigonometric polynomial

$$p(\mathbf{y}) = \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}} e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

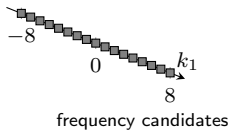
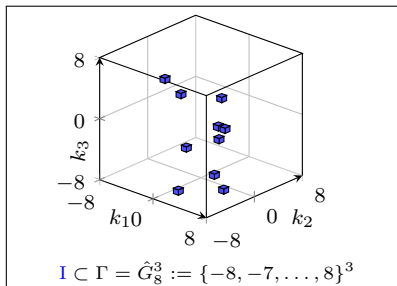
from sampling values.

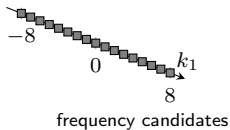
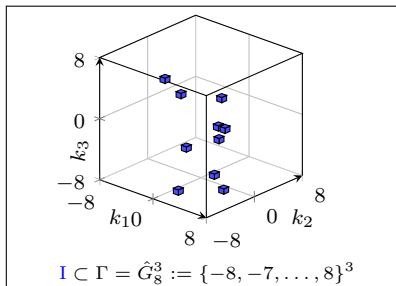
- ▶ Function approximation of  $u(\mathbf{x}_0, \cdot)$  can be realized using the sFFT by assuming it to be a trigonometric polynomial  $p$  with some noise  $\eta$ , i.e.,

$$u(\mathbf{x}_0, \mathbf{y}) = p(\mathbf{y}) + \eta(\mathbf{y}).$$

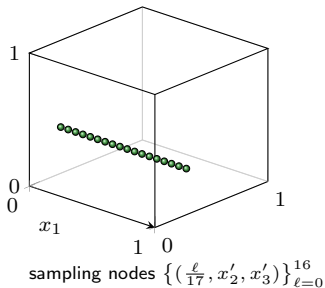


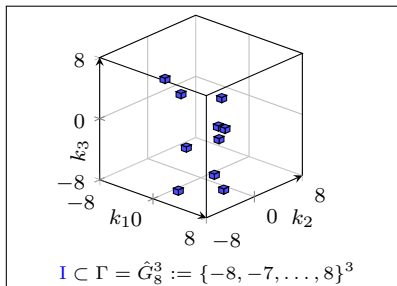
$$I \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \dots, 8\}^3$$





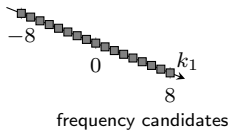
construct  
→  
sampling set



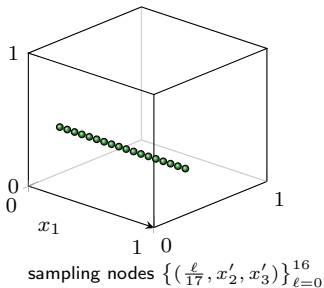


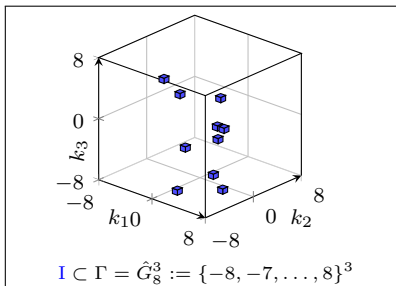
$$\hat{p}_{k_1} := \frac{1}{17} \sum_{\ell=0}^{16} p \left( \begin{pmatrix} \ell/17 \\ x'_2 \\ x'_3 \end{pmatrix} \right) e^{-2\pi i \frac{\ell k_1}{17}}$$

$$k_1 = -8, \dots, 8$$



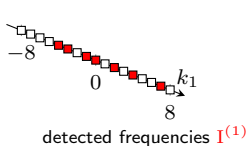
1-dim.  
←  
FFT



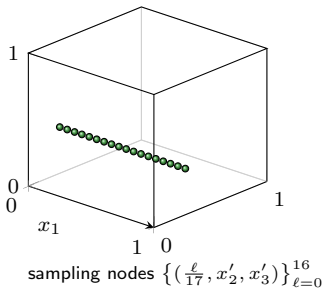


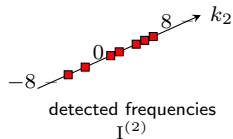
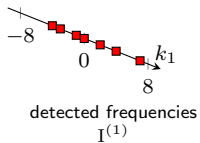
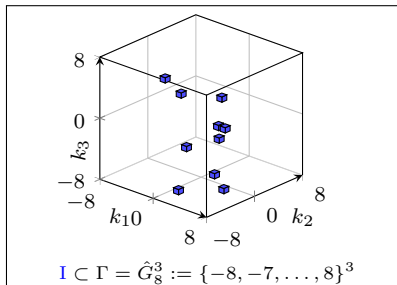
$$\begin{aligned} \hat{p}_{k_1} &:= \frac{1}{17} \sum_{\ell=0}^{16} p \left( \begin{pmatrix} \ell/17 \\ x'_2 \\ x'_3 \end{pmatrix} \right) e^{-2\pi i \frac{\ell k_1}{17}} \\ &= \sum_{\substack{(h_2, h_3) \in \{-8, \dots, 8\}^2 \\ (k_1, h_2, h_3)^\top \in \text{supp } \hat{p}}} \hat{p} \begin{pmatrix} k_1 \\ h_2 \\ h_3 \end{pmatrix} e^{2\pi i (h_2 x'_2 + h_3 x'_3)}, \end{aligned}$$

$$k_1 = -8, \dots, 8$$

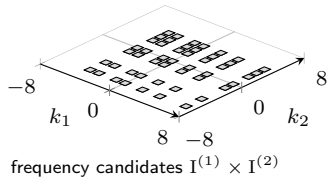
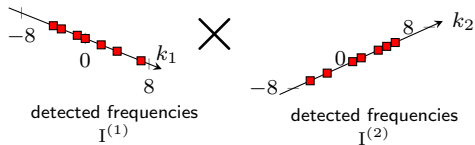
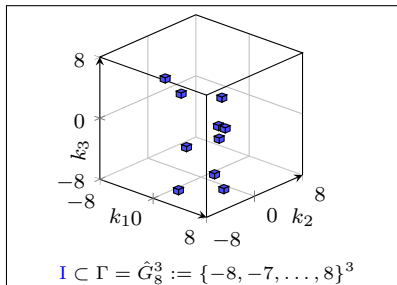


1-dim.  
←  
FFT











► Result:

$$u_{\mathbf{x}_0}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_0}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_0}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

▶ Result:

$$u_{\mathbf{x}_0}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_0}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_0}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

▶ Problems:

- ▶ We are probably interested in more than one spatial node  $\mathbf{x}_0$ .
- ▶ We have to call the sFFT one time for each  $\mathbf{x}_g$ ,  $g = 1, \dots, G$ , to compute

$$u_{\mathbf{x}_g}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_g}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_g}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}} \quad \forall g = 1, \dots, G.$$

► Result:

$$u_{\mathbf{x}_0}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_0}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_0}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

► Problems:

- We are probably interested in more than one spatial node  $\mathbf{x}_0$ .
- We have to call the sFFT one time for each  $\mathbf{x}_g$ ,  $g = 1, \dots, G$ , to compute

$$u_{\mathbf{x}_g}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_g}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_g}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}} \quad \forall g = 1, \dots, G.$$

- But sampling is very expensive! One sample  $u(\mathbf{x}_g, \mathbf{y}) \hat{=}$  one call of the FEM PDE solver for fixed  $\mathbf{y}$  (+ evaluation at  $\mathbf{x}_g$ ).
- Sampling nodes  $\mathbf{y}$  differ for each  $\mathbf{x}_g \rightarrow$  we can't reuse the PDE solutions.

► Result:

$$u_{\mathbf{x}_0}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_0}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_0}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

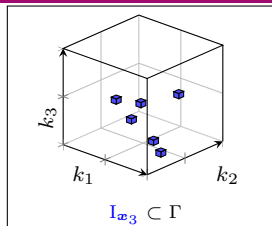
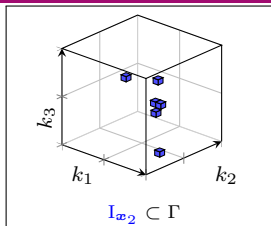
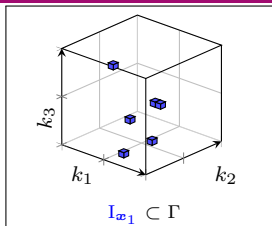
► Problems:

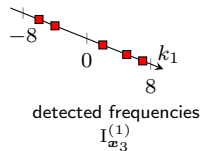
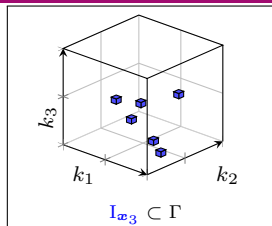
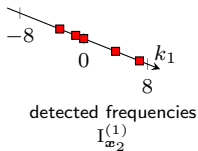
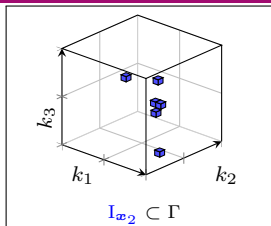
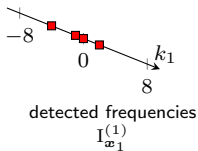
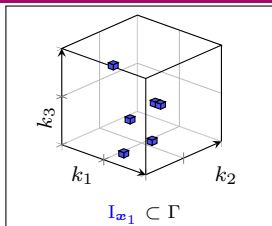
- We are probably interested in more than one spatial node  $\mathbf{x}_0$ .
- We have to call the sFFT one time for each  $\mathbf{x}_g$ ,  $g = 1, \dots, G$ , to compute

$$u_{\mathbf{x}_g}^{\text{sFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in \mathbf{I}_{\mathbf{x}_g}} c_{\mathbf{k}}^{\text{sFFT}}(u_{\mathbf{x}_g}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}} \quad \forall g = 1, \dots, G.$$

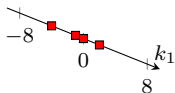
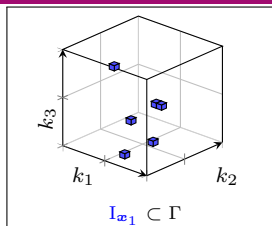
- But sampling is very expensive! One sample  $u(\mathbf{x}_g, \mathbf{y}) \hat{=}$  one call of the FEM PDE solver for fixed  $\mathbf{y}$  (+ evaluation at  $\mathbf{x}_g$ ).
- Sampling nodes  $\mathbf{y}$  differ for each  $\mathbf{x}_g \rightarrow$  we can't reuse the PDE solutions.

► Solution: the *uniform* sFFT!

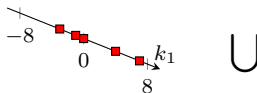
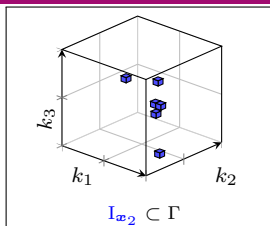




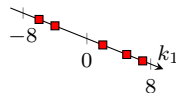
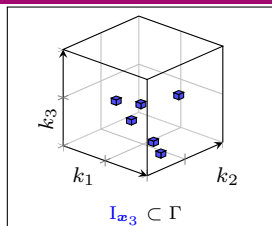




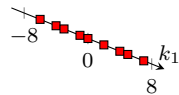
$$I_{x_1}^{(1)}$$



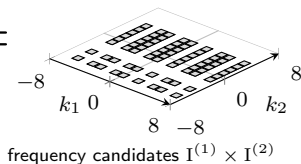
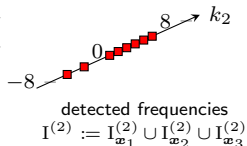
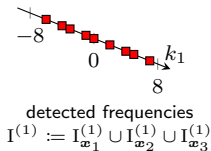
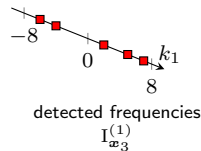
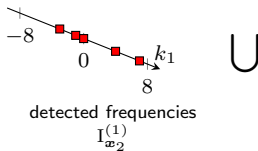
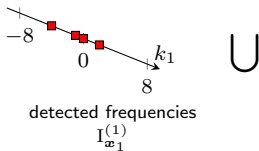
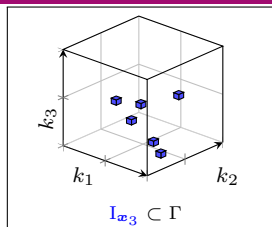
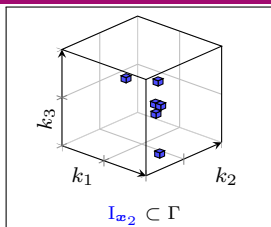
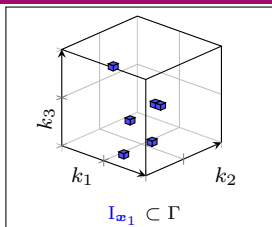
$$I_{x_2}^{(1)}$$

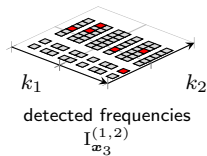
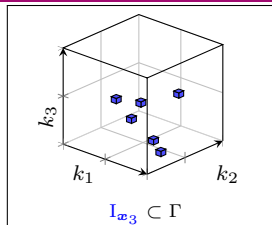
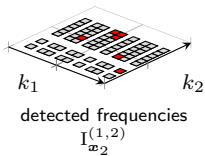
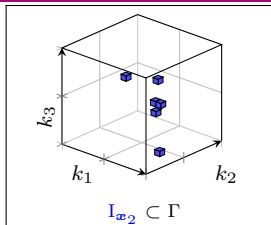
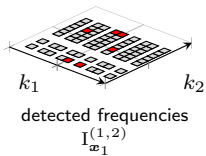
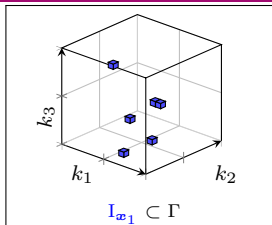


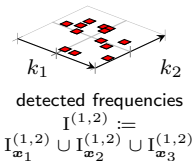
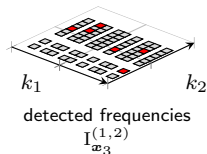
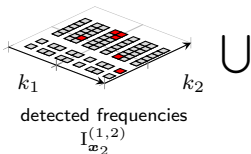
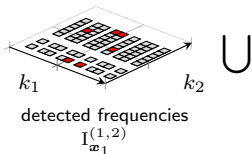
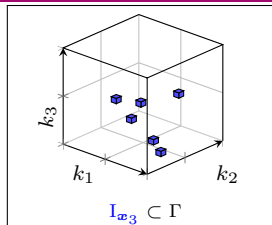
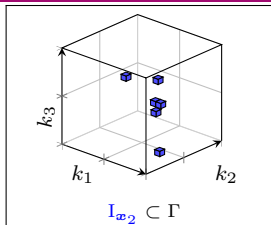
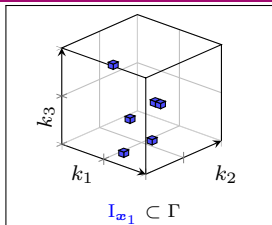
$$I_{x_3}^{(1)}$$

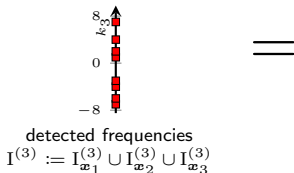
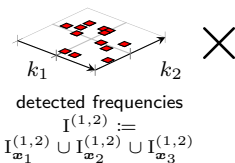
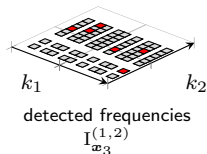
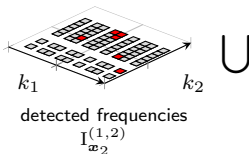
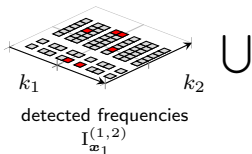
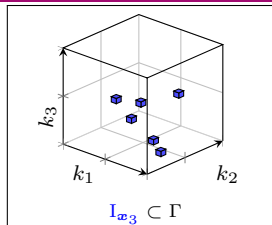
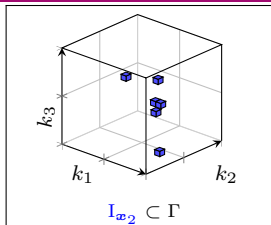
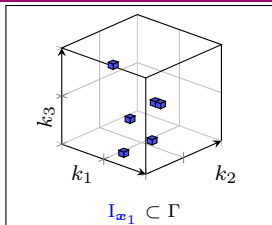


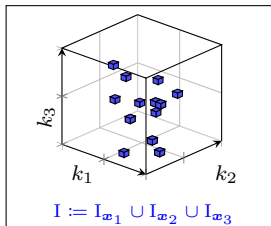
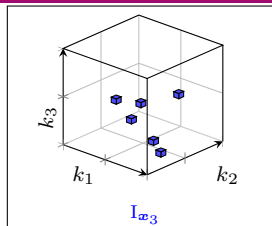
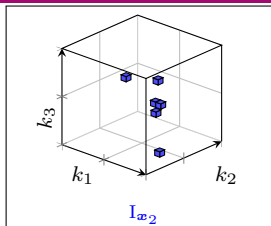
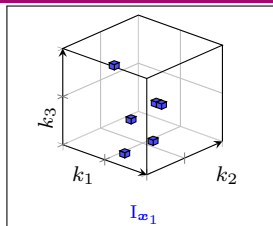
$$I^{(1)} := I_{x_1}^{(1)} \cup I_{x_2}^{(1)} \cup I_{x_3}^{(1)}$$











► Result:

$$u_{x_g}^{\text{usFFT}}(\mathbf{y}) := \sum_{\mathbf{k} \in I} c_{\mathbf{k}}^{\text{usFFT}}(u_{x_g}) e^{2\pi i \mathbf{k} \cdot \mathbf{y}}$$

► Output:

- index set  $I \in \Gamma$  with  $I_{x_g} \subset I$  for all  $g = 1, \dots, G$ , i.e., containing the frequencies  $\mathbf{k}$  corresponding to the sparsity  $s$  approximately largest Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for each  $g = 1, \dots, G$
- approximations  $c_{\mathbf{k}}^{\text{usFFT}}(u_{x_g})$ ,  $\mathbf{k} \in I$ , of the Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for all  $g = 1, \dots, G$

► Output:

- index set  $I \in \Gamma$  with  $I_{x_g} \subset I$  for all  $g = 1, \dots, G$ , i.e., containing the frequencies  $\mathbf{k}$  corresponding to the sparsity  $s$  approximately largest Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for each  $g = 1, \dots, G$
- approximations  $c_{\mathbf{k}}^{\text{usFFT}}(u_{x_g}), \mathbf{k} \in I$ , of the Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for all  $g = 1, \dots, G$

► cardinality of the set  $S$  of sampling locations:

$$|S| \in \mathcal{O} \left( d s \max(s, N_{\Gamma}) \log^2 \frac{d s G N_{\Gamma}}{\delta} + \max(sG, N_{\Gamma}) \log \frac{d s G}{\delta} \right)$$



► Output:

- index set  $I \in \Gamma$  with  $I_{x_g} \subset I$  for all  $g = 1, \dots, G$ , i.e., containing the frequencies  $\mathbf{k}$  corresponding to the sparsity  $s$  approximately largest Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for each  $g = 1, \dots, G$
- approximations  $c_{\mathbf{k}}^{\text{usFFT}}(u_{x_g})$ ,  $\mathbf{k} \in I$ , of the Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for all  $g = 1, \dots, G$

► cardinality of the set  $S$  of sampling locations:

$$|S| \in \mathcal{O} \left( d s \max(s, N_{\Gamma}) \log^2 \frac{d s G N_{\Gamma}}{\delta} + \max(sG, N_{\Gamma}) \log \frac{d s G}{\delta} \right)$$

► computational complexity of the usFFT:

$$\mathcal{O} \left( d^2 s^2 G^2 N_{\Gamma} \log^3 \frac{d s G N_{\Gamma}}{\delta} \right)$$

with probability at least  $1 - \delta$  as well as

$$\mathcal{O} \left( d^2 s^3 G^2 N_{\Gamma} \log^3 \frac{d s G N_{\Gamma}}{\delta} \right)$$

in the worst case

► Output:

- index set  $I \in \Gamma$  with  $I_{x_g} \subset I$  for all  $g = 1, \dots, G$ , i.e., containing the frequencies  $\mathbf{k}$  corresponding to the sparsity  $s$  approximately largest Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for each  $g = 1, \dots, G$
- approximations  $c_{\mathbf{k}}^{\text{usFFT}}(u_{x_g})$ ,  $\mathbf{k} \in I$ , of the Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for all  $g = 1, \dots, G$

► cardinality of the set  $S$  of sampling locations:

$$|S| \in \mathcal{O} \left( d s \max(s, N_{\Gamma}) \log^2 \frac{d s G N_{\Gamma}}{\delta} + \max(s G, N_{\Gamma}) \log \frac{d s G}{\delta} \right)$$

► computational complexity of the usFFT:

$$\mathcal{O} \left( d^2 s^2 G^2 N_{\Gamma} \log^3 \frac{d s G N_{\Gamma}}{\delta} \right)$$

with probability at least  $1 - \delta$  as well as

$$\mathcal{O} \left( d^2 s^3 G^2 N_{\Gamma} \log^3 \frac{d s G N_{\Gamma}}{\delta} \right)$$

in the worst case

► Output:

- index set  $I \in \Gamma$  with  $I_{x_g} \subset I$  for all  $g = 1, \dots, G$ , i.e., containing the frequencies  $\mathbf{k}$  corresponding to the sparsity  $s$  approximately largest Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for each  $g = 1, \dots, G$
- approximations  $c_{\mathbf{k}}^{\text{usFFT}}(u_{x_g})$ ,  $\mathbf{k} \in I$ , of the Fourier coefficients  $c_{\mathbf{k}}(u_{x_g})$  for all  $g = 1, \dots, G$

- cardinality of the set  $S$  of sampling locations:  $G \lesssim ds$

$$|S| \in \mathcal{O} \left( ds \max(s, N_{\Gamma}) \log^2 \frac{dsGN_{\Gamma}}{\delta} + \max(sG, N_{\Gamma}) \log \frac{dsG}{\delta} \right)$$

- computational complexity of the usFFT:

$$\mathcal{O} \left( d^2 s^2 G^2 N_{\Gamma} \log^3 \frac{dsGN_{\Gamma}}{\delta} \right)$$

with probability at least  $1 - \delta$  as well as

$$\mathcal{O} \left( d^2 s^3 G^2 N_{\Gamma} \log^3 \frac{dsGN_{\Gamma}}{\delta} \right)$$

in the worst case

Consider the problem

$$\begin{aligned}
 -\nabla_{\mathbf{x}} \cdot (a(\mathbf{x}, \mathbf{y}) \nabla_{\mathbf{x}} u(\mathbf{x}, \mathbf{y})) &= 1 & \mathbf{x} \in D, \mathbf{y} \in D_{\mathbf{y}} \\
 u(\mathbf{x}, \mathbf{y}) &= 0 & \forall \mathbf{x} \in \partial D, \mathbf{y} \in D_{\mathbf{y}}
 \end{aligned}$$

with  $D = (0, 1)^2$ ,  $\mathbf{y} \sim \mathcal{U}([-1, 1]^{20})$  and the random coefficient

$$a(\mathbf{x}, \mathbf{y}) = 1 + \sum_{j=1}^{20} y_j \psi_j(\mathbf{x})$$

with

$$\psi_j(\mathbf{x}) := \frac{0.9}{\zeta(2)} j^{-2} \cos(2\pi m_1(j)x_1) \cos(2\pi m_2(j)x_2), \quad \mathbf{x} \in D, j \geq 1.$$

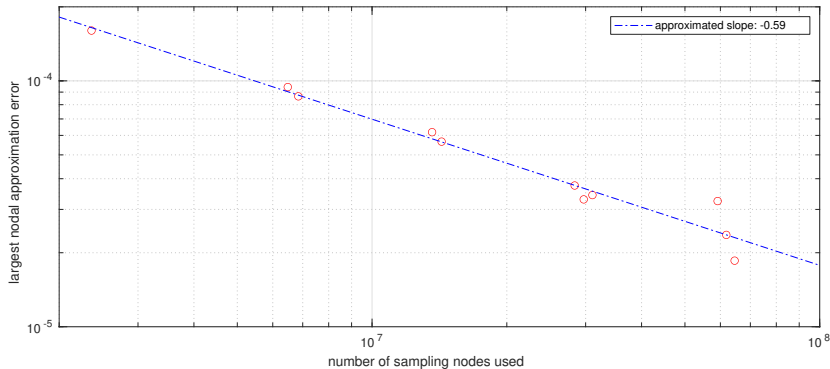
$j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
$m_1(j)$	0	1	0	1	2	0	1	2	3	0	1	2	3	4	...
$m_2(j)$	1	0	2	1	0	3	2	1	0	4	3	2	1	0	...

Example taken from

M. Eigel, C. J. Gittelsohn, C. Schwab and E. Zander.

*Adaptive stochastic Galerkin FEM.*

Comput. Methods Appl. Mech. Engrg., 270: 247–269, 2014.



**Figure:** Largest error  $\text{err}_2^\eta$  w.r.t. the nodes  $\mathbf{x}_g$  for different parameter settings, i.e.,  $s = 100, 250, 500, 1000, 2000$ , for the affine example.

$$\text{err}_2^\eta(\mathbf{x}_g) := \sqrt{\frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} |\tilde{u}(\mathbf{x}_g, \mathbf{y}^{(j)}) - u^{\text{usFFT}}(\mathbf{x}_g, \mathbf{y}^{(j)})|^2}$$

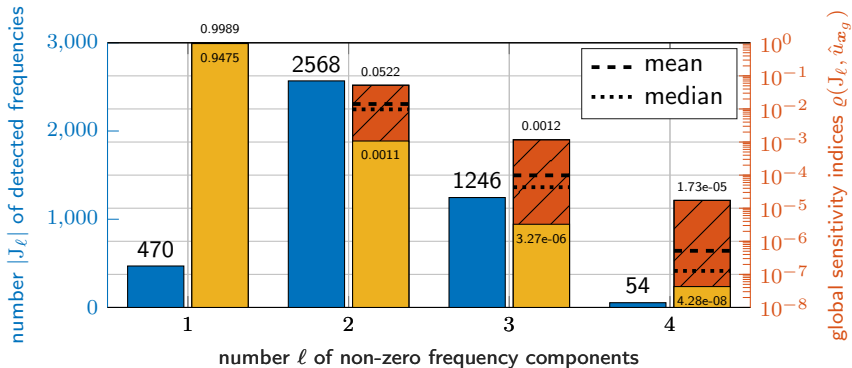


Figure: Analysis of the approximation for the affine example with  $s = 2000$ ,  $N = 32$ .

$$\varrho(J, \tilde{u}_{x_g}^{\text{usFFT}}) := \frac{\sigma^2(\tilde{u}_{x_g, J}^{\text{usFFT}})}{\sigma^2(\tilde{u}_{x_g}^{\text{usFFT}})} = \frac{\sum_{\mathbf{k} \in J \setminus \{\mathbf{0}\}} |c_{\mathbf{k}}^{\text{usFFT}}(\tilde{u}_{x_g})|^2}{\sum_{\mathbf{k} \in I \setminus \{\mathbf{0}\}} |c_{\mathbf{k}}^{\text{usFFT}}(\tilde{u}_{x_g})|^2} \in [0, 1],$$

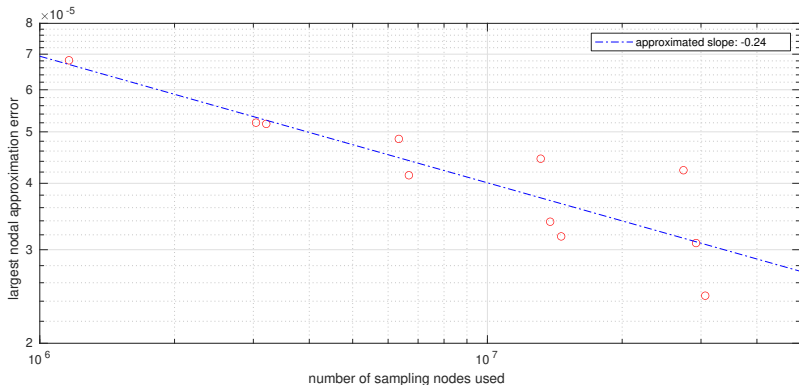
Consider the problem

$$\begin{aligned}
 -\nabla_{\mathbf{x}} \cdot (a(\mathbf{x}, \mathbf{y}) \nabla_{\mathbf{x}} u(\mathbf{x}, \mathbf{y})) &= f(\mathbf{x}) & \mathbf{x} \in D, \mathbf{y} \in D_{\mathbf{y}} \\
 u(\mathbf{x}, \mathbf{y}) &= 0 & \forall \mathbf{x} \in \partial D, \mathbf{y} \in D_{\mathbf{y}}
 \end{aligned}$$

with  $f(\mathbf{x}) = \sin(1.3\pi x_1 + 3.4\pi x_2) \cos(4.3\pi x_1 - 3.1\pi x_2)$ ,  $D = (0, 1)^2$ ,  $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and the random coefficient

$$\log a(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{10} \frac{y_j}{j} \sin(2\pi j x_1) \cos(2\pi(11-j)x_2).$$

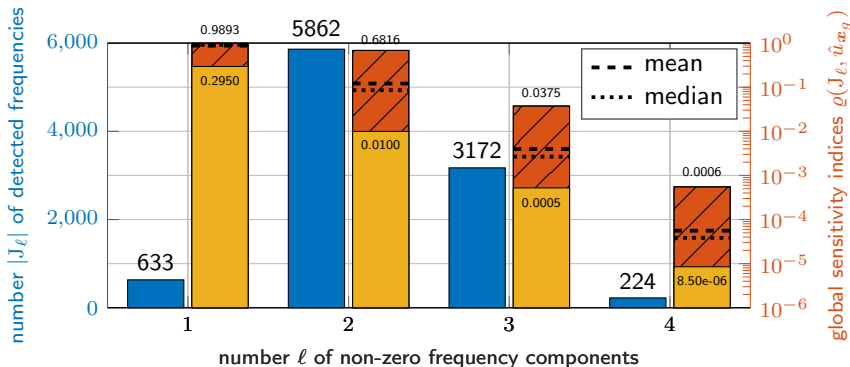
Example taken and modified from  
 M. Cheng, T. Y. Hou, M. Yan and Z. Zhang.  
*A data-driven stochastic method for elliptic PDEs with random coefficients.*  
 SIAM/ASA J. Uncertain. Quantif., 1: 452-493, 2013.



**Figure:** Largest error  $\text{err}_2^\eta$  w.r.t. the nodes  $\mathbf{x}_g$  for different parameter settings, i.e.,  $s = 100, 250, 500, 1000, 2000$ , for the lognormal example.

$$\text{err}_2^\eta(\mathbf{x}_g) := \sqrt{\frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} |\tilde{u}(\mathbf{x}_g, \mathbf{y}^{(j)}) - u^{\text{usFFT}}(\mathbf{x}_g, \mathbf{y}^{(j)})|^2}$$





**Figure:** Analysis of the approximation for the lognormal example with  $s = 2000$ ,  $N = 32$ .

$$\varrho(J, \tilde{u}_{x_g}^{\text{usFFT}}) := \frac{\sigma^2(\tilde{u}_{x_g, J}^{\text{usFFT}})}{\sigma^2(\tilde{u}_{x_g}^{\text{usFFT}})} = \frac{\sum_{\mathbf{k} \in J \setminus \{\mathbf{0}\}} |c_{\mathbf{k}}^{\text{usFFT}}(\tilde{u}_{x_g})|^2}{\sum_{\mathbf{k} \in I \setminus \{\mathbf{0}\}} |c_{\mathbf{k}}^{\text{usFFT}}(\tilde{u}_{x_g})|^2} \in [0, 1],$$

- ▶ Main advantages of the usFFT:
  - ▶ fully adaptive, no critical a priori choice needed
  - ▶ sample efficient (in terms of sampling locations)
  - ▶ approximation gives insight on the influence and interactions of the  $y_j$
  - ▶ non-intrusive and parallelizable
  
- ▶ Lutz Kämmerer, Daniel Potts, Fabian Taubert  
*The uniform sparse FFT with application to PDEs with random coefficients*  
ArXiv e-prints, 2021. arXiv:2109.04131 [math.NA]
  
- ▶ Thank you for your attention!