The uniform sparse FFT

with application to PDEs with random coefficients

Fabian Taubert

joint work with Lutz Kämmerer and Daniel Potts

Chemnitz University of Technology
Applied Functional Analysis

Curves and Surfaces 2022
21.06.2022

Outline

General aim

Approximation (by using samples) of the solution $u(\boldsymbol{x}, \boldsymbol{y})$ of the PDE

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

spatial variable $\boldsymbol{x}$

$\boldsymbol{x} \in D \subset \mathbb{R}^{d_{\boldsymbol{x}}}$, $d_{\boldsymbol{x}} \in \{1, 2, 3\}$

random variable $\boldsymbol{y}$

$\boldsymbol{y} = (y_j)_{j=1}^{d} \in D_{\boldsymbol{y}}$, d very large

Affine random coefficient

$\boldsymbol{y}$ typically uniformly distributed in $D_{\boldsymbol{y}}$, e.g. $\boldsymbol{y} \sim \mathcal{U}\left([-1,1]^d\right)$

$$a(\boldsymbol{x},\boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j=1}^{d} y_j\,\psi_j(\boldsymbol{x})$$

[Cohen, DeVore, Schwab '10], [Dick, Kuo, Le Gia, Schwab '16], [Bachmayr, Cohen, Dahmen '18], [Gantner, Herrmann, Schwab '18], ...

Lognormal random coefficient

$\boldsymbol{y}$ typically normally distributed in $D_{\boldsymbol{y}} = \mathbb{R}^d$, i.e. $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$

$$a(\boldsymbol{x},\boldsymbol{y}) = a_0(\boldsymbol{x}) + \exp(b(\boldsymbol{x},\boldsymbol{y})), \qquad b(\boldsymbol{x},\boldsymbol{y}) = b_0(\boldsymbol{x}) + \sum_{j=1}^{d} y_j\,\psi_j(\boldsymbol{x})$$

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

Affine random coefficient

$\boldsymbol{y}$ typically uniformly distributed in $D_{\boldsymbol{y}}$, e.g. $\boldsymbol{y} \sim \mathcal{U}\left([-1,1]^d\right)$

$$a(\boldsymbol{x},\boldsymbol{y}) = a_0(\boldsymbol{x}) + \sum_{j=1}^{d} y_j\,\psi_j(\boldsymbol{x})$$

[Cohen, DeVore, Schwab '10], [Dick, Kuo, Le Gia, Schwab '16], [Bachmayr, Cohen, Dahmen '18], [Gantner, Herrmann, Schwab '18], ...

Lognormal random coefficient

$\boldsymbol{y}$ typically normally distributed in $D_{\boldsymbol{y}} = \mathbb{R}^d$, i.e. $\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{0},\boldsymbol{I}\right)$

$$a(\boldsymbol{x},\boldsymbol{y}) = a_0(\boldsymbol{x}) + \exp(b(\boldsymbol{x},\boldsymbol{y})), \qquad b(\boldsymbol{x},\boldsymbol{y}) = b_0(\boldsymbol{x}) + \sum_{j=1}^{d} y_j\,\psi_j(\boldsymbol{x})$$

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

General aim

Approximation of the solution $u(\boldsymbol{x}, \boldsymbol{y})$

Main problem

d typically very large
$\Longrightarrow$ Curse of Dimensionality

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!
Also, they often just approximate some quantity of interest (e.g. $\mathbb{E}[F(u(\cdot, \boldsymbol{y}))]$).

General aim

Approximation of the solution $u(\boldsymbol{x}, \boldsymbol{y})$

Main problem

d typically very large
$\Longrightarrow$ Curse of Dimensionality

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!
Also, they often just approximate some quantity of interest (e.g. $\mathbb{E}[F(u(\cdot, \boldsymbol{y}))]$).

General aim

Approximation of the solution $u(\boldsymbol{x}, \boldsymbol{y})$

Main problem

d typically very large
$\implies$ Curse of Dimensionality

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!
Also, they often just approximate some quantity of interest (e.g. $\mathbb{E}[F(u(\cdot, \boldsymbol{y}))]$).

Main Tool

a sparse FFT (sFFT) approach [Indyk, Kapralov '12], [Potts, Volkmer '16]

The sFFT was originally designed to recover sparse trigonometric polynomials

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

with unknown frequency set $\mathrm{I} \subset \mathbb{Z}^d$.

Input

- search space $\Gamma \subset \mathbb{Z}^d$
- sparsity $s \geq |\mathrm{I}|$
- black box sampling $(\boldsymbol{y} \mapsto p(\boldsymbol{y}))$

Output

- frequency set $\mathrm{I} = \operatorname{supp} \hat{p} \subset \Gamma \subset \mathbb{Z}^d$
- coefficients $\hat{p}_{\boldsymbol{k}}, \boldsymbol{k} \in \mathrm{I}$

Function approximation using the sFFT is realized via thresholding.

Main Tool

a sparse FFT (sFFT) approach [Indyk, Kapralov '12], [Potts, Volkmer '16]

The sFFT was originally designed to recover sparse trigonometric polynomials

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

with unknown frequency set $\mathrm{I} \subset \mathbb{Z}^d$.

Input

- search space $\Gamma \subset \mathbb{Z}^d$
- sparsity $s \geq |\mathrm{I}|$
- black box sampling $(\boldsymbol{y} \mapsto p(\boldsymbol{y}))$

Output

- frequency set $\mathrm{I} = \operatorname{supp} \hat{p} \subset \Gamma \subset \mathbb{Z}^d$
- coefficients $\hat{p}_{\boldsymbol{k}}, \boldsymbol{k} \in \mathrm{I}$

Function approximation using the sFFT is realized via thresholding.

Main Tool

a sparse FFT (sFFT) approach [Indyk, Kapralov '12], [Potts, Volkmer '16]

The sFFT was originally designed to recover sparse trigonometric polynomials

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

with unknown frequency set $\mathrm{I} \subset \mathbb{Z}^d$.

Input

- search space $\Gamma \subset \mathbb{Z}^d$
- sparsity $s \geq |\mathrm{I}|$
- black box sampling $(\boldsymbol{y} \mapsto p(\boldsymbol{y}))$

Output

- frequency set $\mathrm{I} = \operatorname{supp} \hat{p} \subset \Gamma \subset \mathbb{Z}^d$
- coefficients $\hat{p}_{\boldsymbol{k}}, \boldsymbol{k} \in \mathrm{I}$

Function approximation using the sFFT is realized via thresholding.

Main Tool

a sparse FFT (sFFT) approach [Indyk, Kapralov '12], [Potts, Volkmer '16]

The sFFT was originally designed to recover sparse trigonometric polynomials

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

with unknown frequency set $\mathrm{I} \subset \mathbb{Z}^d$.

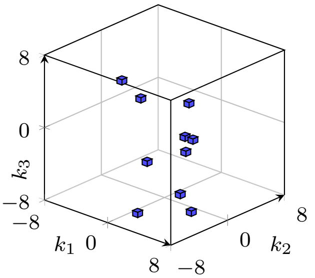
Input

- search space $\Gamma \subset \mathbb{Z}^d$
- sparsity $s \geq |\mathrm{I}|$
- black box sampling $(\boldsymbol{y} \mapsto p(\boldsymbol{y}))$

Output

- frequency set $\mathrm{I} = \operatorname{supp} \hat{p} \subset \Gamma \subset \mathbb{Z}^d$
- coefficients $\hat{p}_{\boldsymbol{k}}, \boldsymbol{k} \in \mathrm{I}$

Function approximation using the sFFT is realized via thresholding.



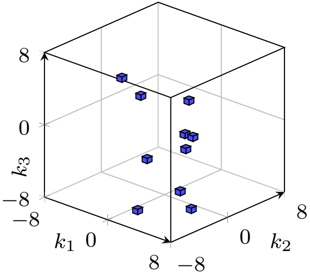
$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$

$$\hat{p}_{k_1} := \frac{1}{17} \sum_{\ell=0}^{16} p\left(\begin{pmatrix} \ell/17 \\ x_2' \\ x_3' \end{pmatrix}\right) \mathrm{e}^{-2\pi\mathrm{i}\frac{\ell k_1}{17}}$$

$k_1 = -8, \ldots, 8$

frequency candidates

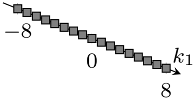
sampling nodes $\{(\frac{\ell}{17}, x_2', x_3')\}_{\ell=0}^{16}$



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

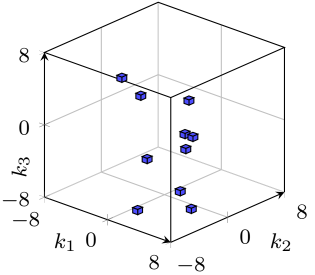
$$\hat{p}_{k_1} := \frac{1}{17} \sum_{\ell=0}^{16} p\left(\begin{pmatrix} \ell/17 \\ x_2' \\ x_3' \end{pmatrix}\right) \mathrm{e}^{-2\pi \mathrm{i} \frac{\ell k_1}{17}}$$

$k_1 = -8, \ldots, 8$



frequency candidates

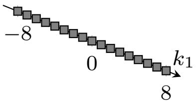
sampling nodes $\{(\frac{\ell}{17}, x_2', x_3')\}_{\ell=0}^{16}$



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

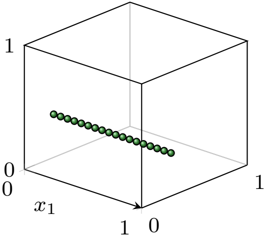
$$\hat{p}_{k_1} := \frac{1}{17} \sum_{\ell=0}^{16} p\left(\begin{pmatrix} \ell/17 \\ x_2' \\ x_3' \end{pmatrix}\right) \mathrm{e}^{-2\pi\mathrm{i}\frac{\ell k_1}{17}}$$

$k_1 = -8, \ldots, 8$

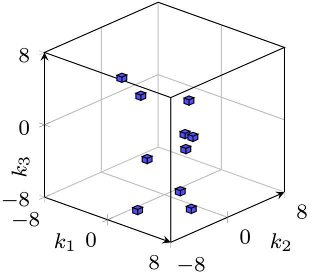


frequency candidates

construct $\longrightarrow$ sampling set



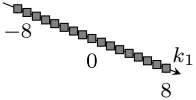
sampling nodes $\{(\frac{\ell}{17}, x_2', x_3')\}_{\ell=0}^{16}$



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

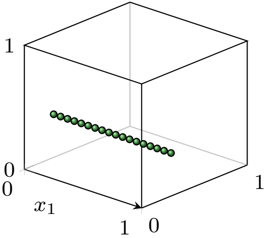
$$\hat{p}_{k_1} := \frac{1}{17} \sum_{\ell=0}^{16} p\left(\begin{pmatrix} \ell/17 \\ x_2' \\ x_3' \end{pmatrix}\right) \mathrm{e}^{-2\pi \mathrm{i} \frac{\ell k_1}{17}}$$

$k_1 = -8, \dots, 8$

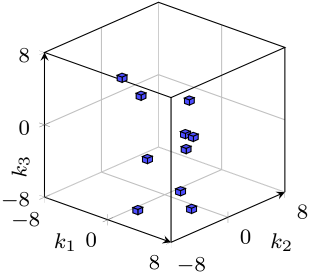


frequency candidates

1-dim. ⟵ FFT



sampling nodes $\{(\frac{\ell}{17}, x_2', x_3')\}_{\ell=0}^{16}$

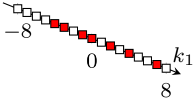


$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$

$$\hat{p}_{k_1} := \frac{1}{17} \sum_{\ell=0}^{16} p\left(\begin{pmatrix} \ell/17 \\ x_2' \\ x_3' \end{pmatrix}\right) \mathrm{e}^{-2\pi\mathrm{i}\frac{\ell k_1}{17}}$$

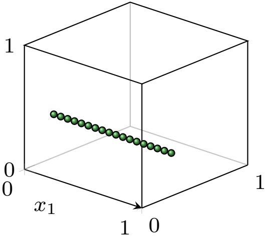
$$= \sum_{\substack{(h_2,h_3)\in\{-8,\ldots,8\}^2 \\ (k_1,h_2,h_3)^\top \in \operatorname{supp}\hat{p}}} \hat{p}_{\begin{pmatrix} k_1 \\ h_2 \\ h_3 \end{pmatrix}} \mathrm{e}^{2\pi\mathrm{i}(h_2 x_2' + h_3 x_3')},$$

$k_1 = -8, \ldots, 8$

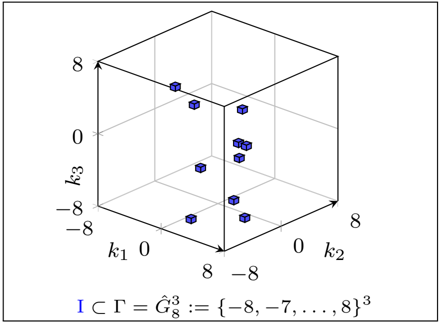


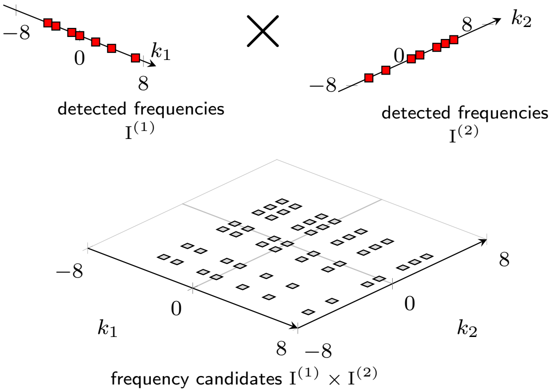
detected frequencies $\mathrm{I}^{(1)}$

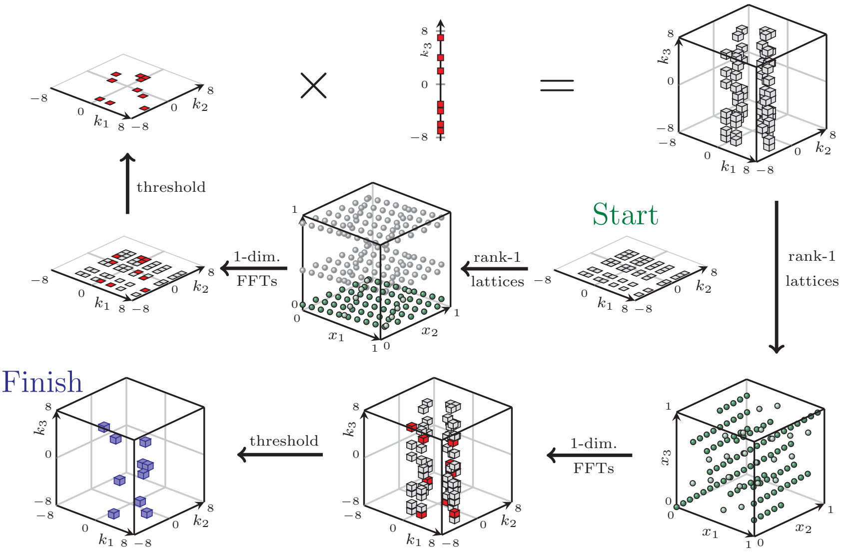
1-dim.
$\longleftarrow$
FFT



sampling nodes $\left\{\left(\frac{\ell}{17}, x_2', x_3'\right)\right\}_{\ell=0}^{16}$


8
0
k_3
−8
−8
k_1 0
8 −8
0 k_2
8
$\mathrm{I} \subset \Gamma = \hat{G}_8^3 := \{-8, -7, \ldots, 8\}^3$


−8
0
k_1
8
detected frequencies
$\mathrm{I}^{(1)}$
k_2
8
0
−8
detected frequencies
$\mathrm{I}^{(2)}$
−8
0
k_1
8 −8
0
k_2
8
frequency candidates $\mathrm{I}^{(1)} \times \mathrm{I}^{(2)}$


Start
rank-1 lattices
1-dim. FFTs
threshold
×
=
rank-1 lattices
1-dim. FFTs
threshold
Finish

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

$$u^{\mathrm{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathrm{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}}$$

Problem: Amount of samples

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

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

$$u^{\mathrm{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathrm{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}}$$

Problem: Amount of samples

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

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

$$u^{\mathrm{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathrm{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}}$$

Problem: Amount of samples

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

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

$$u^{\mathtt{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathtt{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}}$$

Problem: Amount of samples

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

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

$$u^{\mathrm{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathrm{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}}$$

Problem: Amount of samples

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

$u(\boldsymbol{x}_1, \cdot)$	different functions	$u(\boldsymbol{x}_2, \cdot)$
$\downarrow$		$\downarrow$
$\mathrm{I}_{\boldsymbol{x}_1}$	different index sets	$\mathrm{I}_{\boldsymbol{x}_2}$
$\downarrow$		$\downarrow$
$K_{\boldsymbol{x}_1}$	different candidate sets	$K_{\boldsymbol{x}_2}$
$\downarrow$		$\downarrow$
$S_{\boldsymbol{x}_1}$	different sampling nodes	$S_{\boldsymbol{x}_2}$

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

$$u^{\mathrm{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathrm{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{y}}$$

Problem: Amount of samples

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

$u(\boldsymbol{x}_1, \cdot)$	slightly different functions	$u(\boldsymbol{x}_2, \cdot)$
↓ $\mathrm{I}_{\boldsymbol{x}_1}$	slightly different index sets	↓ $\mathrm{I}_{\boldsymbol{x}_2}$
↓ $K_{\boldsymbol{x}_1}$	slightly different candidate sets	↓ $K_{\boldsymbol{x}_2}$
↓ $S_{\boldsymbol{x}_1}$	different sampling nodes	↓ $S_{\boldsymbol{x}_2}$

Aim

Find an approximation of $u(\boldsymbol{x}, \boldsymbol{y})$.

Spatial discretization

Fix $\boldsymbol{x}_g \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_G\}$ and consider the d-variate function $u(\boldsymbol{x}_g, \cdot)$.

Adaptive approximation at each $\boldsymbol{x}_g$

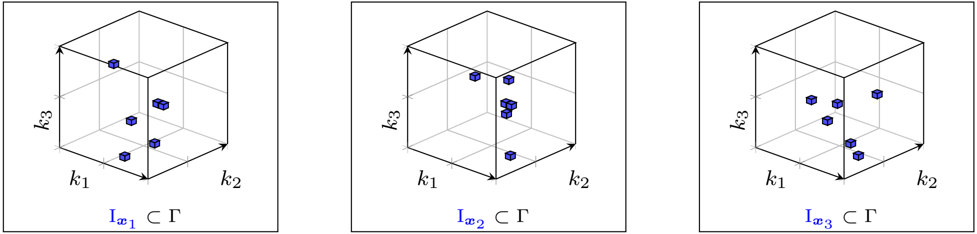
$$u^{\mathrm{sFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}_{\boldsymbol{x}_g}} c^{\mathrm{sFFT}}_{\boldsymbol{k}, \boldsymbol{x}_g} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}}$$

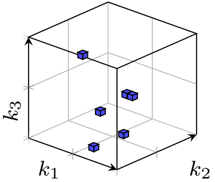
Problem: Amount of samples

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

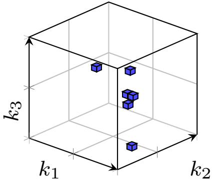
$u(\boldsymbol{x}_1, \cdot)$	slightly different functions	$u(\boldsymbol{x}_2, \cdot)$
$\downarrow$		$\downarrow$
$\mathrm{I}_{\boldsymbol{x}_1}$	slightly different index sets	$\mathrm{I}_{\boldsymbol{x}_2}$
$\downarrow$		$\downarrow$
$K_{\boldsymbol{x}_1}$	slightly different candidate sets	$K_{\boldsymbol{x}_2}$
$\downarrow$		$\downarrow$
$S_{\boldsymbol{x}_1}$	different sampling nodes	$S_{\boldsymbol{x}_2}$

- How to utilize this?
- Solution: the uniform sFFT (usFFT)

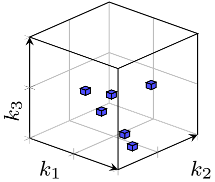

k_3
k_1
k_2
$\mathrm{I}_{\boldsymbol{x}_1} \subset \Gamma$
k_3
k_1
k_2
$\mathrm{I}_{\boldsymbol{x}_2} \subset \Gamma$
k_3
k_1
k_2
$\mathrm{I}_{\boldsymbol{x}_3} \subset \Gamma$



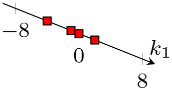
$\mathrm{I}_{\boldsymbol{x}_1} \subset \Gamma$



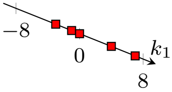
$\mathrm{I}_{\boldsymbol{x}_2} \subset \Gamma$



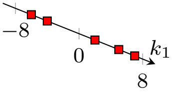
$\mathrm{I}_{\boldsymbol{x}_3} \subset \Gamma$



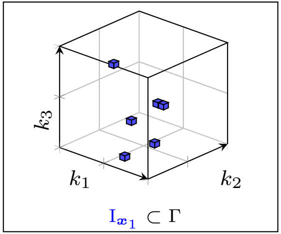
detected frequencies $\mathrm{I}_{\boldsymbol{x}_1}^{(1)}$

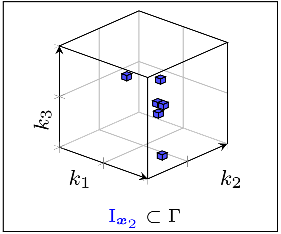


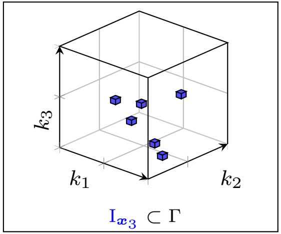
detected frequencies $\mathrm{I}_{\boldsymbol{x}_2}^{(1)}$

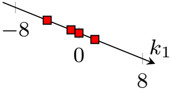


detected frequencies $\mathrm{I}_{\boldsymbol{x}_3}^{(1)}$



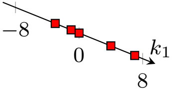






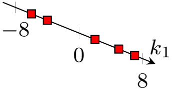
detected frequencies
$I^{(1)}_{\boldsymbol{x}_1}$

$\bigcup$

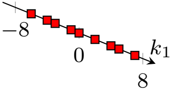


detected frequencies
$I^{(1)}_{\boldsymbol{x}_2}$

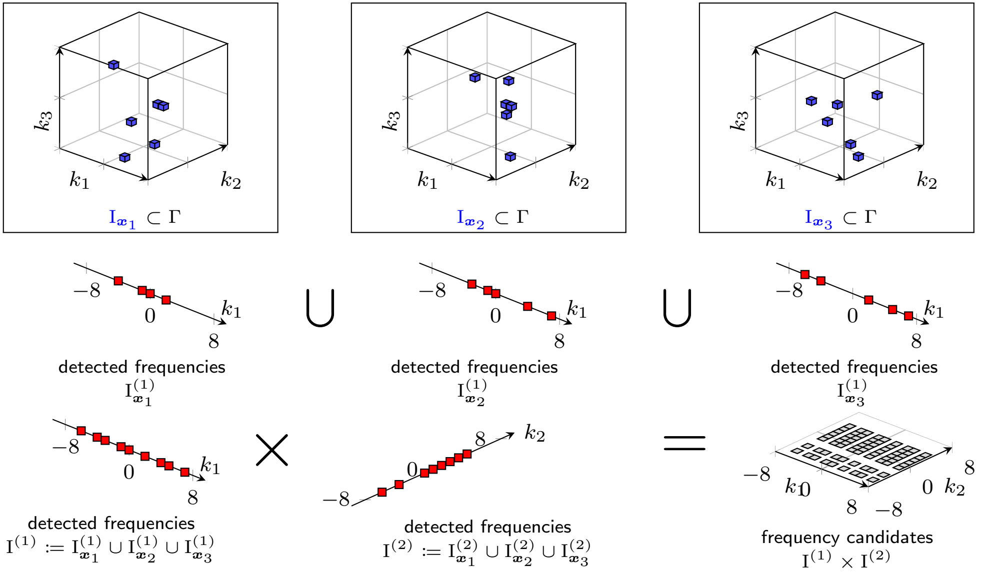
$\bigcup$

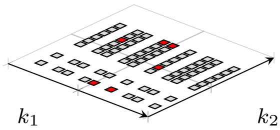


detected frequencies
$I^{(1)}_{\boldsymbol{x}_3}$

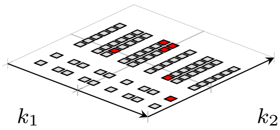


detected frequencies
$I^{(1)} := I^{(1)}_{\boldsymbol{x}_1} \cup I^{(1)}_{\boldsymbol{x}_2} \cup I^{(1)}_{\boldsymbol{x}_3}$

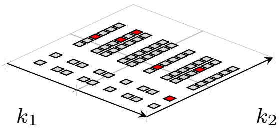

k_3
k_1
k_2
$I_{x_1} \subset \Gamma$
$I_{x_2} \subset \Gamma$
$I_{x_3} \subset \Gamma$
−8
0
8
$\cup$
detected frequencies
$I^{(1)}_{x_1}$
$I^{(1)}_{x_2}$
$I^{(1)}_{x_3}$
$\times$
=
detected frequencies
$I^{(1)} := I^{(1)}_{x_1} \cup I^{(1)}_{x_2} \cup I^{(1)}_{x_3}$
detected frequencies
$I^{(2)} := I^{(2)}_{x_1} \cup I^{(2)}_{x_2} \cup I^{(2)}_{x_3}$
frequency candidates
$I^{(1)} \times I^{(2)}$



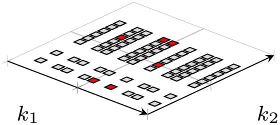
detected frequencies
$\mathrm{I}_{\boldsymbol{x}_1}^{(1,2)}$



detected frequencies
$\mathrm{I}_{\boldsymbol{x}_2}^{(1,2)}$

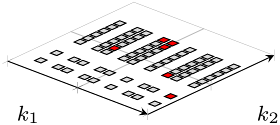


detected frequencies
$\mathrm{I}_{\boldsymbol{x}_3}^{(1,2)}$



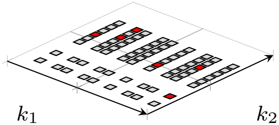
detected frequencies
$\mathrm{I}_{\boldsymbol{x}_1}^{(1,2)}$

$\cup$

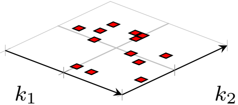


detected frequencies
$\mathrm{I}_{\boldsymbol{x}_2}^{(1,2)}$

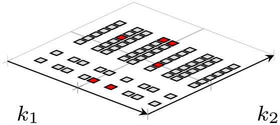
$\cup$



detected frequencies
$\mathrm{I}_{\boldsymbol{x}_3}^{(1,2)}$

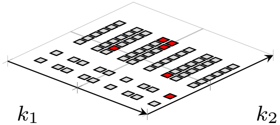


detected frequencies
$\mathrm{I}^{(1,2)} := \mathrm{I}_{\boldsymbol{x}_1}^{(1,2)} \cup \mathrm{I}_{\boldsymbol{x}_2}^{(1,2)} \cup \mathrm{I}_{\boldsymbol{x}_3}^{(1,2)}$



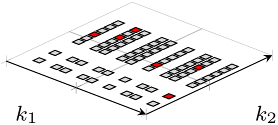
detected frequencies
$\mathrm{I}_{\boldsymbol{x}_1}^{(1,2)}$

$\cup$

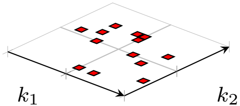


detected frequencies
$\mathrm{I}_{\boldsymbol{x}_2}^{(1,2)}$

$\cup$

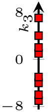


detected frequencies
$\mathrm{I}_{\boldsymbol{x}_3}^{(1,2)}$



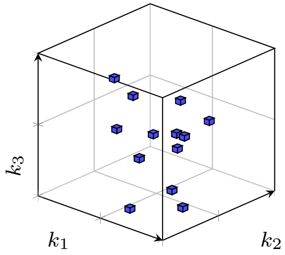
detected frequencies
$\mathrm{I}^{(1,2)} := \mathrm{I}_{\boldsymbol{x}_1}^{(1,2)} \cup \mathrm{I}_{\boldsymbol{x}_2}^{(1,2)} \cup \mathrm{I}_{\boldsymbol{x}_3}^{(1,2)}$

$\times$



detected frequencies
$\mathrm{I}^{(3)} := \mathrm{I}_{\boldsymbol{x}_1}^{(3)} \cup \mathrm{I}_{\boldsymbol{x}_2}^{(3)} \cup \mathrm{I}_{\boldsymbol{x}_3}^{(3)}$

$=$ $\ldots$



$\mathrm{I} := \mathrm{I}_{\boldsymbol{x}_1} \cup \mathrm{I}_{\boldsymbol{x}_2} \cup \mathrm{I}_{\boldsymbol{x}_3}$

- Result:

$$u(\boldsymbol{x}_g, \boldsymbol{y}) \approx u^{\texttt{usFFT}}(\boldsymbol{x}_g, \boldsymbol{y}) := \sum_{\boldsymbol{k} \in \mathrm{I}} c_{\boldsymbol{k}, \boldsymbol{x}_g}^{\texttt{usFFT}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{y}} \qquad g = 1, \ldots, G$$

Output of the usFFT

- index set $\mathrm{I} \subset \Gamma \subset \mathbb{Z}^d$ with $\mathrm{I}_{\boldsymbol{x}_g} \subset \mathrm{I}$ for all $g = 1, \ldots, G$
- approximations $c^{\mathtt{usFFT}}_{\boldsymbol{k},\boldsymbol{x}_g}, \boldsymbol{k} \in \mathrm{I}$, for all $g = 1, \ldots, G$

Sample Complexity (= how often do we have to solve the PDE)

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

Output of the usFFT

- index set $\mathrm{I} \subset \Gamma \subset \mathbb{Z}^d$ with $\mathrm{I}_{\boldsymbol{x}_g} \subset \mathrm{I}$ for all $g = 1, \dots, G$
- approximations $c^{\texttt{usFFT}}_{\boldsymbol{k},\boldsymbol{x}_g}, \boldsymbol{k} \in \mathrm{I}$, for all $g = 1, \dots, G$

Sample Complexity (= how often do we have to solve the PDE)

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

Output of the usFFT

- index set $\mathrm{I} \subset \Gamma \subset \mathbb{Z}^d$ with $\mathrm{I}_{\boldsymbol{x}_g} \subset \mathrm{I}$ for all $g = 1, \dots, G$
- approximations $c^{\mathtt{usFFT}}_{\boldsymbol{k},\boldsymbol{x}_g}, \boldsymbol{k} \in \mathrm{I}$, for all $g = 1, \dots, G$

Sample Complexity (= how often do we have to solve the PDE)

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

Output of the usFFT

- index set $\mathrm{I} \subset \Gamma \subset \mathbb{Z}^d$ with $\mathrm{I}_{\boldsymbol{x}_g} \subset \mathrm{I}$ for all $g = 1, \ldots, G$
- approximations $c^{\mathtt{usFFT}}_{\boldsymbol{k},\boldsymbol{x}_g}, \boldsymbol{k} \in \mathrm{I}$, for all $g = 1, \ldots, G$

Sample Complexity (= how often do we have to solve the PDE)

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

Output of the usFFT

- index set $\mathrm{I} \subset \Gamma \subset \mathbb{Z}^d$ with $\mathrm{I}_{\boldsymbol{x}_g} \subset \mathrm{I}$ for all $g = 1, \dots, G$
- approximations $c_{\boldsymbol{k},\boldsymbol{x}_g}^{\texttt{usFFT}}, \boldsymbol{k} \in \mathrm{I}$, for all $g = 1, \dots, G$

Sample Complexity (= how often do we have to solve the PDE)

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

PDE (Example from [Eigel, Gittelson, Schwab, Zander 14'])

$$-\nabla_{\boldsymbol{x}} \cdot (a(\boldsymbol{x},\boldsymbol{y})\nabla_{\boldsymbol{x}} u(\boldsymbol{x},\boldsymbol{y})) = 1 \qquad \boldsymbol{x} \in D,\ \boldsymbol{y} \in D_{\boldsymbol{y}}$$

$$u(\boldsymbol{x},\boldsymbol{y}) = 0 \qquad \forall \boldsymbol{x} \in \partial D,\ \boldsymbol{y} \in D_{\boldsymbol{y}}$$

Spatial domain

$$D = (0,1)^2$$

Random domain

$$D_{\boldsymbol{y}} = [-1,1]^{20}$$

Distribution

$$\boldsymbol{y} \sim \mathcal{U}\left([-1,1]^{20}\right)$$

Affine random coefficient

$$a(\boldsymbol{x},\boldsymbol{y}) = 1 + \sum_{j=1}^{20} y_j \psi_j(\boldsymbol{x}) \quad \text{with} \quad \psi_j(\boldsymbol{x}) := \frac{0.9}{\zeta(2)} j^{-2} \cos(2\pi m_1(j) x_1) \cos(2\pi m_2(j) x_2)$$

and

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	...

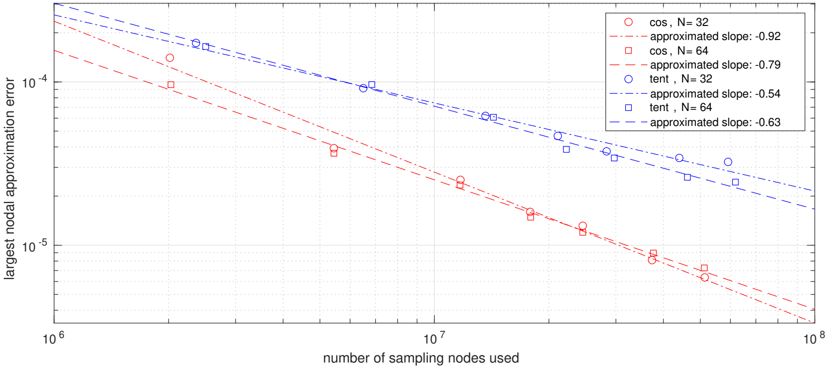


Figure: Largest error err_2^η w.r.t. the nodes $\boldsymbol{x}_g$ for different parameter settings, i.e., $s \in \{100, 250, 500, 750, 1000, 1500, 2000\}$, for the affine example.

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

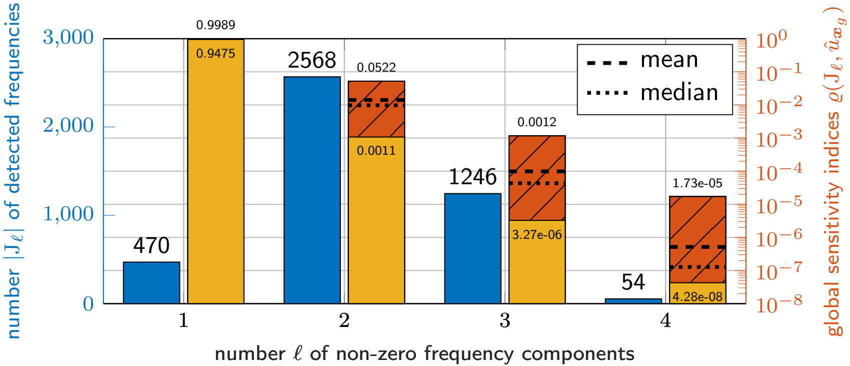


Figure: Analysis of the approximation for the affine example with $s = 2000,\ N = 32$.

$$\varrho(\mathrm{J}, \tilde{u}^{\mathrm{usFFT}}_{\boldsymbol{x}_g}) := \frac{\sigma^2(\tilde{u}^{\mathrm{usFFT}}_{\boldsymbol{x}_g,\mathrm{J}})}{\sigma^2(\tilde{u}^{\mathrm{usFFT}}_{\boldsymbol{x}_g})} = \frac{\sum_{\boldsymbol{k}\in\mathrm{J}\setminus\{\mathbf{0}\}} |c^{\mathrm{usFFT}}_{\boldsymbol{k}}(\tilde{u}_{\boldsymbol{x}_g})|^2}{\sum_{\boldsymbol{k}\in\mathrm{I}\setminus\{\mathbf{0}\}} |c^{\mathrm{usFFT}}_{\boldsymbol{k}}(\tilde{u}_{\boldsymbol{x}_g})|^2} \in [0,1],$$

PDE (Example from [Cheng, Hou, Yan, Zhang 13'], modified)

$$-\nabla_{\boldsymbol{x}} \cdot (a(\boldsymbol{x},\boldsymbol{y})\nabla_{\boldsymbol{x}} u(\boldsymbol{x},\boldsymbol{y})) = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in D,\ \boldsymbol{y} \in D_{\boldsymbol{y}}$$

$$u(\boldsymbol{x},\boldsymbol{y}) = 0 \qquad \forall \boldsymbol{x} \in \partial D,\ \boldsymbol{y} \in D_{\boldsymbol{y}}$$

with $f(\boldsymbol{x}) = \sin(1.3\pi x_1 + 3.4\pi x_2)\cos(4.3\pi x_1 - 3.1\pi x_2)$

Spatial domain

$$D = (0,1)^2$$

Random domain

$$D_{\boldsymbol{y}} = \mathbb{R}^{10}$$

Distribution

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Lognormal random coefficient

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

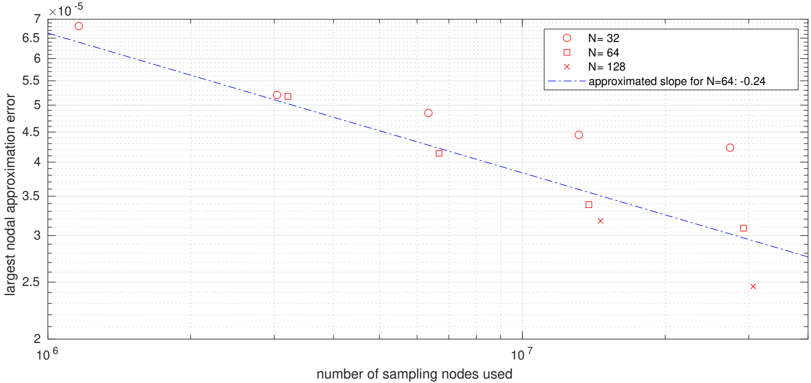


Figure: Largest error err_2^η w.r.t. the nodes $\boldsymbol{x}_g$ for different parameter settings, i.e., $s \in \{100, 250, 500, 1000, 2000\}$, for the lognormal example.

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

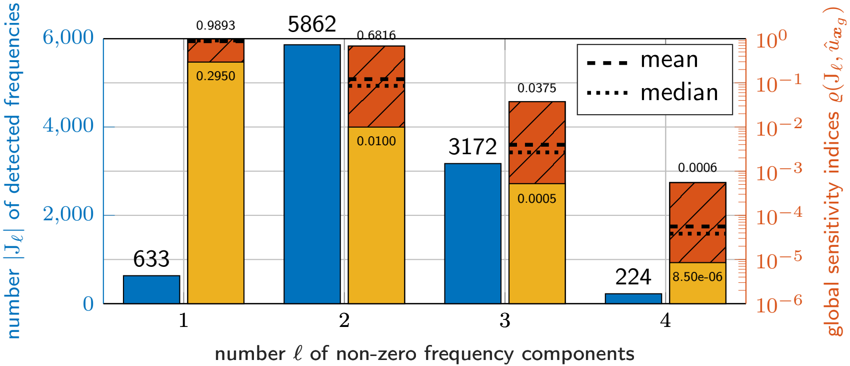


Figure: Analysis of the approximation for the lognormal example with $s = 2000,\ N = 32$.

$$\varrho(\mathrm{J}, \tilde{u}^{\mathrm{usFFT}}_{\boldsymbol{x}_g}) := \frac{\sigma^2(\tilde{u}^{\mathrm{usFFT}}_{\boldsymbol{x}_g,\mathrm{J}})}{\sigma^2(\tilde{u}^{\mathrm{usFFT}}_{\boldsymbol{x}_g})} = \frac{\sum_{\boldsymbol{k}\in\mathrm{J}\setminus\{\boldsymbol{0}\}} |c^{\mathrm{usFFT}}_{\boldsymbol{k}}(\tilde{u}_{\boldsymbol{x}_g})|^2}{\sum_{\boldsymbol{k}\in\mathrm{I}\setminus\{\boldsymbol{0}\}} |c^{\mathrm{usFFT}}_{\boldsymbol{k}}(\tilde{u}_{\boldsymbol{x}_g})|^2} \in [0,1],$$

- 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 (does not affect the PDE solver) and parallelizable
  - adapts easily to other domains, boundary conditions, ...

- 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!

- 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 (does not affect the PDE solver) and parallelizable
  - adapts easily to other domains, boundary conditions, ...

- 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!