Stochastic Collocation for Partial Differential Equations with Random Coefficients

Nicholas Krämer

May 28, 2018

Outline

PDEs with random coefficients

Stochastic collocation

Sparse interpolation operators

Convergence and stability of sparse interpolation

Conclusion

PDEs with random coefficients

What is a PDE with random coefficients?

- **It looks like:**

  Find u such that for almost every $\omega \in \Omega$:

$$\begin{cases} \mathcal{L}(\omega)u = f & \text{on } D \\ \mathcal{B}(\omega)u = g & \text{on } \partial D. \end{cases}$$

What is a PDE with random coefficients?

- **It looks like:**

  Find u such that for almost every $\omega \in \Omega$:

$$\begin{cases} \mathcal{L}(\omega)u = f & \text{on } D \\ \mathcal{B}(\omega)u = g & \text{on } \partial D. \end{cases}$$

- **For illustration purposes we treat:**

  Find u such that for almost every $\omega \in \Omega$:

$$\begin{cases} -\operatorname{div}(a(\cdot,\omega)\nabla u) = f & \text{on } D \\ u = 0 & \text{on } \partial D. \end{cases}$$

Weak formulation

- Find u s.t. for almost all $\omega \in \Omega$ and for all $v \in H_0^1(D)$:

$$\int_D a(x,\omega)\nabla u(x,\omega) \cdot \nabla v(x) \; \mathrm{d}x = \int_D f(x) v(x) \; \mathrm{d}x$$

Weak formulation

- Find u s.t. for almost all $\omega \in \Omega$ and for all $v \in H_0^1(D)$:

$$\int_D a(x,\omega)\nabla u(x,\omega) \cdot \nabla v(x) \, \mathrm{d}x = \int_D f(x)v(x) \, \mathrm{d}x$$

- Well posed if
  - $f \in L^2(D)$
  - $a_{\mathrm{max}}(\omega) \geq a(x,\omega) \geq a_{\mathrm{min}}(\omega) > 0$ almost surely/everywhere

Weak formulation

- Find u s.t. for almost all $\omega \in \Omega$ and for all $v \in H_0^1(D)$:

$$\int_D a(x,\omega)\nabla u(x,\omega) \cdot \nabla v(x) \, \mathrm{d}x = \int_D f(x)v(x) \, \mathrm{d}x$$

- Well posed if
  - $f \in L^2(D)$
  - $a_{\max}(\omega) \geq a(x,\omega) \geq a_{\min}(\omega) > 0$ almost surely/everywhere
- Quantity of interest : $\mathbb{E}[u(x)]$, $\mathrm{Cov}[u(x), u(y)]$, ...

Weak formulation

- Find u s.t. for almost all $\omega \in \Omega$ and for all $v \in H_0^1(D)$:

$$\int_D a(x,\omega)\nabla u(x,\omega)\cdot\nabla v(x)\ \mathrm{d}x = \int_D f(x)v(x)\ \mathrm{d}x$$

- Well posed if
  - $f \in L^2(D)$
  - $a_{\max}(\omega) \geq a(x,\omega) \geq a_{\min}(\omega) > 0$ almost surely/everywhere
- Quantity of interest : $\mathbb{E}[u(x)]$, $\mathrm{Cov}[u(x), u(y)]$, ...
- Monte Carlo methods or **spectral approximation** ($\rightarrow$ SC)

Spectral approximation

- Reduce random field to **countable** collection of r.v.:

$$a(x,\omega) = \sum_{n\in\mathbb{N}} \varphi_n(x) Y_n(\omega)$$

(Karhunen-Loève, Polynomial Chaos, ...)

Spectral approximation

- Reduce random field to **countable** collection of r.v.:

$$a(x,\omega) = \sum_{n\in\mathbb{N}} \varphi_n(x) Y_n(\omega)$$

(Karhunen-Loève, Polynomial Chaos, ...)

- Replace Ω by $(Y_1(\Omega), Y_2(\Omega), \ldots) \subseteq \mathbb{R}^{\mathbb{N}}$

Spectral approximation

- Reduce random field to **countable** collection of r.v.:

$$a(x,\omega) = \sum_{n\in\mathbb{N}} \varphi_n(x) Y_n(\omega)$$

  (Karhunen-Loève, Polynomial Chaos, ...)
- Replace Ω by $(Y_1(\Omega), Y_2(\Omega), \ldots) \subseteq \mathbb{R}^{\mathbb{N}}$
- Approximate the mapping

$$\begin{cases} \mathbb{R}^{\mathbb{N}} \longrightarrow H_0^1(D) \\ y \longmapsto u(y) = u(a(y)) \end{cases}$$

Spectral approximation

- Reduce random field to **countable** collection of r.v.:

$$a(x,\omega) = \sum_{n\in\mathbb{N}} \varphi_n(x) Y_n(\omega)$$

  (Karhunen-Loève, Polynomial Chaos, ...)
- Replace Ω by $(Y_1(\Omega), Y_2(\Omega), \dots) \subseteq \mathbb{R}^{\mathbb{N}}$
- Approximate the mapping

$$\begin{cases} \mathbb{R}^{\mathbb{N}} \longrightarrow H_0^1(D) \\ y \longmapsto u(y) = u(a(y)) \end{cases}$$

- **Can be a very high-dimensional problem!**

Stochastic collocation

(A special form of spectral approximation)

Breaking down the problem

- Finite noise assumption (not necessary but illustrative):

$$a(x,\omega) = a(x, Y_1(\omega), Y_2(\omega))$$

Breaking down the problem

- Finite noise assumption (not necessary but illustrative):

$$a(x,\omega) = a(x, Y_1(\omega), Y_2(\omega))$$

- **Problem to solve:**
  Find u s.t. for all $v \in H_0^1(D)$:

$$\int_D a(x,y_1,y_2)\nabla u(x,y_1,y_2)\cdot\nabla v(x)\ \mathrm{d}x = \int_D f(x)v(x)\ \mathrm{d}x$$

  for almost all $(y_1,y_2) \in \Sigma := Y_1(\Omega)\times Y_2(\Omega) \subseteq \mathbb{R}^2$

Breaking down the problem

- Finite noise assumption (not necessary but illustrative):

$$a(x,\omega) = a(x, Y_1(\omega), Y_2(\omega))$$

- **Problem to solve:**

  Find u s.t. for all $v \in H_0^1(D)$:

$$\int_D a(x, y_1, y_2)\nabla u(x, y_1, y_2) \cdot \nabla v(x)\, \mathrm{d}x = \int_D f(x)v(x)\, \mathrm{d}x$$

  for almost all $(y_1, y_2) \in \Sigma := Y_1(\Omega) \times Y_2(\Omega) \subseteq \mathbb{R}^2$

- For single evaluations $\gamma = (\gamma_1, \gamma_2)$ we can solve/approx. for u_γ

Stochastic collocation in a nutshell

- Solve the problem for a lot of evaluations on some grid

Stochastic collocation in a nutshell

- Solve the problem for a lot of evaluations on some grid
- Interpolate in the Σ-variable (the "random" input)

Stochastic collocation in a nutshell

- Solve the problem for a lot of evaluations on some grid
- Interpolate in the Σ-variable (the "random" input)
- "Collocation" is actually "interpolation"

Stochastic collocation in a nutshell

- Solve the problem for a lot of evaluations on some grid
- Interpolate in the Σ-variable (the “random” input)
- “Collocation” is actually “interpolation”
- SC is **vector-valued interpolation** in the stochastic variable

Stochastic collocation in a nutshell

- Solve the problem for a lot of evaluations on some grid
- Interpolate in the Σ-variable (the "random" input)
- "Collocation" is actually "interpolation"
- SC is **vector-valued interpolation** in the stochastic variable

Again:

Solve the problem for each gridpoint $\rightarrow$ interpolate

Stochastic collocation in a nutshell

- Solve the problem for a lot of evaluations on some grid
- Interpolate in the Σ-variable (the “random” input)
- “Collocation” is actually “interpolation”
- SC is **vector-valued interpolation** in the stochastic variable

Again:

Solve the problem for each gridpoint $\rightarrow$ interpolate

Important:

Also works for countably many variables $\rightarrow$ **sparse** interpolation

Polynomial interpolation 1/2

- Work in each direction ($i = 1, 2$)

Polynomial interpolation 1/2

- Work in each direction ($i = 1, 2$), $\Sigma = \Sigma_1 \otimes \Sigma_2$

Polynomial interpolation 1/2

- Work in each direction ($i = 1, 2$), $\Sigma = \Sigma_1 \otimes \Sigma_2$
- Discretize Σ_i with a grid: $\Gamma_i := \{t_{0,i}, ..., t_{N_i,i}\}$

Polynomial interpolation 1/2

- Work in each direction ($i = 1, 2$), $\Sigma = \Sigma_1 \otimes \Sigma_2$
- Discretize Σ_i with a grid: $\Gamma_i := \{t_{0,i}, ..., t_{N_i,i}\}$
- Lagrange basis: $\ell_{j,i}(t_{k,i}) = \delta_{jk}$ and interpolation operator

$$\mathcal{I}_i v(t) := \sum_{j=1}^{N_i} \ell_{j,i}(t) v(t_j)$$

Polynomial interpolation 1/2

- Work in each direction ($i = 1, 2$), $\Sigma = \Sigma_1 \otimes \Sigma_2$
- Discretize Σ_i with a grid: $\Gamma_i := \{t_{0,i}, ..., t_{N_i,i}\}$
- Lagrange basis: $\ell_{j,i}(t_{k,i}) = \delta_{jk}$ and interpolation operator

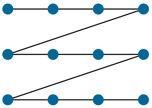
$$\mathcal{I}_i v(t) := \sum_{j=1}^{N_i} \ell_{j,i}(t) v(t_j)$$

- Interpolation space $\mathcal{P}_{N_i}(\Sigma_i)$ corresponding to $\mathcal{I}_i$ and Γ_i

Polynomial interpolation 2/2

- Global enumeration of local indices:

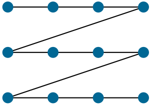
$$k := k_1 + N_1 k_2 \in \{0, ..., \underbrace{(N_1+1)(N_2+1)}_{=:N_P}\}$$



Polynomial interpolation 2/2

- Global enumeration of local indices:

$$k := k_1 + N_1 k_2 \in \{0, \dots, \underbrace{(N_1+1)(N_2+1)}_{=:N_P}\}$$

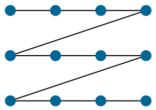


- Tensor-Lagrange basis $\ell_k(t_1, t_2) := \ell_{k_1,1}(t_1)\ell_{k_2,2}(t_2)$

Polynomial interpolation 2/2

- Global enumeration of local indices:

$$k := k_1 + N_1 k_2 \in \{0, \dots, \underbrace{(N_1+1)(N_2+1)}_{=:N_P}\}$$



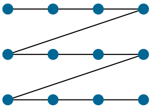
- Tensor-Lagrange basis $\ell_k(t_1, t_2) := \ell_{k_1,1}(t_1)\ell_{k_2,2}(t_2)$
- Interpolation operator

$$\mathcal{I}_\Sigma v(t,s) := \sum_{k=1}^{N_P} \ell_k(t,s) v(t_{k_1,1}, t_{k_2,2})$$

Polynomial interpolation 2/2

- Global enumeration of local indices:

$$k := k_1 + N_1 k_2 \in \{0, \dots, \underbrace{(N_1+1)(N_2+1)}_{=:N_P}\}$$



- Tensor-Lagrange basis $\ell_k(t_1, t_2) := \ell_{k_1,1}(t_1)\ell_{k_2,2}(t_2)$
- Interpolation operator

$$\mathcal{I}_\Sigma v(t,s) := \sum_{k=1}^{N_P} \ell_k(t,s) v(t_{k_1,1}, t_{k_2,2})$$

- Interpolation space $\mathcal{P}_{(N_1,N_2)} := \mathcal{P}_{N_1}(\Sigma_1) \otimes \mathcal{P}_{N_2}(\Sigma_2)$

Putting it together

- Solve equation **at each gridpoint** with FEM:

$$u_{h,k}(x) \approx u(x, t_{k_1,1}, t_{k_2,2})$$

Putting it together

- Solve equation **at each gridpoint** with FEM:

$$u_{h,k}(x) \approx u(x, t_{k_1,1}, t_{k_2,2})$$

- Interpolate these solutions for final approximation

$$u(x, y_1, y_2) \approx u_{h,p}(x, y_1, y_2) = \sum_{k=1}^{N_P} \ell_k(y_1, y_k) u_{h,k}(x)$$

Putting it together

- Solve equation **at each gridpoint** with FEM:

$$u_{h,k}(x) \approx u(x, t_{k_1,1}, t_{k_2,2})$$

- Interpolate these solutions for final approximation

$$u(x, y_1, y_2) \approx u_{h,p}(x, y_1, y_2) = \sum_{k=1}^{N_P} \ell_k(y_1, y_k) u_{h,k}(x)$$

Error sources:

Interpolation error + approximation error (e.g. FEM)

Putting it together

- Solve equation **at each gridpoint** with FEM:

$$u_{h,k}(x) \approx u(x, t_{k_1,1}, t_{k_2,2})$$

- Interpolate these solutions for final approximation

$$u(x, y_1, y_2) \approx u_{h,p}(x, y_1, y_2) = \sum_{k=1}^{N_P} \ell_k(y_1, y_k) u_{h,k}(x)$$

Error sources (think of: "variance-bias decomposition"):

Interpolation error + approximation error (e.g. FEM)

What does the more general case look like?

Computing a quantity of interest

- Assume joint probability density function ρ for the r.v.:

$$\mathbb{E}u(x) = \int_{\Omega} u(x,\omega)\ \mathrm{d}\mathbb{P}(\omega) \approx \int_{\Sigma} u_{h,p}(x,y)\rho(y)\ \mathrm{d}y$$

Computing a quantity of interest

- Assume joint probability density function ρ for the r.v.:

$$\mathbb{E}u(x) = \int_\Omega u(x,\omega)\, \mathrm{d}\mathbb{P}(\omega) \approx \int_\Sigma u_{h,p}(x,y)\rho(y)\, \mathrm{d}y$$

- Introduce auxiliary density $\hat{\rho}$ for weighted Gaussian quadrature

$$\int_\Sigma u_{h,p}(x,y)\rho(y)\, \mathrm{d}y = \sum_{k=1}^{N_P} u_{h,k}(x) \int_\Sigma \ell_k(y) \frac{\rho(y)}{\hat{\rho}(y)} \hat{\rho}(y)\, \mathrm{d}y$$

Computing a quantity of interest

- Assume joint probability density function ρ for the r.v.:

$$\mathbb{E}u(x) = \int_\Omega u(x,\omega)\ \mathrm{d}\mathbb{P}(\omega) \approx \int_\Sigma u_{h,p}(x,y)\rho(y)\ \mathrm{d}y$$

- Introduce auxiliary density $\hat{\rho}$ for weighted Gaussian quadrature

$$\int_\Sigma u_{h,p}(x,y)\rho(y)\ \mathrm{d}y = \sum_{k=1}^{N_P} u_{h,k}(x) \int_\Sigma \ell_k(y)\frac{\rho(y)}{\hat{\rho}(y)}\hat{\rho}(y)\ \mathrm{d}y$$

- Compute (possibly high-dimensional) quadrature

$$\int_\Sigma \ell_k(y)\frac{\rho(y)}{\hat{\rho}(y)}\hat{\rho}(y)\ \mathrm{d}y \approx \sum_{i=1}^{N_Q} w_i \ell_k(x_i)\frac{\rho(x_i)}{\hat{\rho}(x_i)}$$

Comparison to stochastic Galerkin

Stochastic Galerkin:

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$
- Fully coupled system of linear equations possible

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$
- Fully coupled system of linear equations possible

Stochastic Collocation:

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$
- Fully coupled system of linear equations possible

Stochastic Collocation:

- Decoupled system of linear equations

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$
- Fully coupled system of linear equations possible

Stochastic Collocation:

- Decoupled system of linear equations
- Deal with nonindependent r.v. or nonlinear equations

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$
- Fully coupled system of linear equations possible

Stochastic Collocation:

- Decoupled system of linear equations
- Deal with nonindependent r.v. or nonlinear equations

Analogy between SC and SG:

Comparison to stochastic Galerkin

Stochastic Galerkin:

- Ansatz and testfunctions for weak form. from $\mathcal{P}_N(\Sigma) \otimes V_h$
- Fully coupled system of linear equations possible

Stochastic Collocation:

- Decoupled system of linear equations
- Deal with nonindependent r.v. or nonlinear equations

Analogy between SC and SG:

- For certain interpolation points they give the same result

Summary

1. Expand random field $\rightarrow$ countably many random variables

Summary

1. Expand random field $\rightarrow$ countably many random variables
2. Truncate that expansion (finite noise) or not

Summary

1. Expand random field $\rightarrow$ countably many random variables
2. Truncate that expansion (finite noise) or not
3. Solve PDE for points on tensor product grid

Summary

1. Expand random field $\rightarrow$ countably many random variables
2. Truncate that expansion (finite noise) or not
3. Solve PDE for points on tensor product grid
4. Interpolate on that grid to obtain an approximation

Summary

1. Expand random field $\rightarrow$ countably many random variables
2. Truncate that expansion (finite noise) or not
3. Solve PDE for points on tensor product grid
4. Interpolate on that grid to obtain an approximation
5. (High-dimensional grid - curse of dimensionality)

Summary

1. Expand random field $\rightarrow$ countably many random variables
2. Truncate that expansion (finite noise) or not
3. Solve PDE for points on tensor product grid
4. Interpolate on that grid to obtain an approximation
5. (High-dimensional grid - curse of dimensionality)
6. SC decouples system of equations

A shorter version of that summary

1. Finite or countable noise

A shorter version of that summary

1. Finite or countable noise
2. Solve PDE for points on tensor product grid

A shorter version of that summary

1. Finite or countable noise
2. Solve PDE for points on tensor product grid
3. Interpolate on that grid to obtain an approximation

Sparse interpolation operators

Construction of sparse operators

- Abstract point of view: u solves parametric PDE

$$\mathcal{G}(a, u) = 0, \quad \mathcal{G} : X \times V \longrightarrow W$$

Construction of sparse operators

- Abstract point of view: u solves parametric PDE

$$\mathcal{G}(a, u) = 0, \quad \mathcal{G} : X \times V \longrightarrow W$$

- No finite noise: a depends on countably many r.v., $y = (y_1, ...)$

Construction of sparse operators

- Abstract point of view: u solves parametric PDE

$$\mathcal{G}(a, u) = 0, \quad \mathcal{G} : X \times V \longrightarrow W$$

- No finite noise: a depends on countably many r.v., $y = (y_1, \ldots)$
- Work in $[-1,1]^{\mathbb{N}}$; interpolate mapping

$$\begin{cases} [-1,1]^{\mathbb{N}} \longrightarrow V \\ y \longmapsto u(y) \end{cases}$$

Construction of sparse operators

- Abstract point of view: u solves parametric PDE

$$\mathcal{G}(a, u) = 0, \quad \mathcal{G} : X \times V \longrightarrow W$$

- No finite noise: a depends on countably many r.v., $y = (y_1, ...)$
- Work in $[-1, 1]^{\mathbb{N}}$; interpolate mapping

$$\begin{cases} [-1, 1]^{\mathbb{N}} \longrightarrow V \\ y \longmapsto u(y) \end{cases}$$

- Approach: interpolation tools in $[-1, 1] \rightarrow [-1, 1]^{\mathbb{N}} \rightarrow$ sparse

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k \in \mathbb{N}} \subseteq [-1, 1]$ sequence of points

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k\in\mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k \in \mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation
  - Grid: $\Gamma_k := \{t_0, ..., t_k\}$ (nested)

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k \in \mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation
  - Grid: $\Gamma_k := \{t_0, ..., t_k\}$ (nested)
  - Operator: $\mathcal{I}_k$ on Γ_k

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k \in \mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation
  - Grid: $\Gamma_k := \{t_0, ..., t_k\}$ (nested)
  - Operator: $\mathcal{I}_k$ on Γ_k
  - Space: $\mathcal{P}_k([-1, 1])$

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k\in\mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation
  - Grid: $\Gamma_k := \{t_0, ..., t_k\}$ (nested)
  - Operator: $\mathcal{I}_k$ on Γ_k
  - Space: $\mathcal{P}_k([-1, 1])$
  - Difference operator $\Delta_k := \mathcal{I}_k - \mathcal{I}_{k-1}$, $\mathcal{I}_{-1} = 0$

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k\in\mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation
  - Grid: $\Gamma_k := \{t_0, ..., t_k\}$ (nested)
  - Operator: $\mathcal{I}_k$ on Γ_k
  - Space: $\mathcal{P}_k([-1, 1])$
  - Difference operator $\Delta_k := \mathcal{I}_k - \mathcal{I}_{k-1}$, $\mathcal{I}_{-1} = 0$
- Nestnedness: $\Delta_k u(t) = \alpha_k h_k(t)$, $\alpha_k := u(t_k) - I_{k-1}u(t_k)$

Interpolation in $[-1, 1]$

- Let $T := (t_k)_{k \in \mathbb{N}} \subseteq [-1, 1]$ sequence of points
- Define the interpolation
  - Grid: $\Gamma_k := \{t_0, ..., t_k\}$ (nested)
  - Operator: $\mathcal{I}_k$ on Γ_k
  - Space: $\mathcal{P}_k([-1, 1])$
  - Difference operator $\Delta_k := \mathcal{I}_k - \mathcal{I}_{k-1}$, $\mathcal{I}_{-1} = 0$
- Nestnedness: $\Delta_k u(t) = \alpha_k h_k(t)$, $\alpha_k := u(t_k) - I_{k-1} u(t_k)$
- "Hierarchical basis": $\{h_0, ..., h_k\}$ for $\mathcal{P}_k$

Interpolation in infinite dimensions 1/2

- Use as an index-set if working on $\mathbb{R}^{\mathbb{N}}$:

$$\mathscr{F} := \{\nu \in \ell^{\infty}(\mathbb{N}) : \|\nu\|_0 < \infty\}, \quad \|\nu\|_0 = |\{j \in \mathbb{N} : \ \nu_j \neq 0\}|$$

Interpolation in infinite dimensions 1/2

- Use as an index-set if working on $\mathbb{R}^{\mathbb{N}}$:

  $$\mathscr{F} := \{\nu \in \ell^\infty(\mathbb{N}) : \|\nu\|_0 < \infty\}, \quad \|\nu\|_0 = |\{j \in \mathbb{N} : \ \nu_j \neq 0\}|$$

- An interpolation operator is well-defined if it uniquely defines a certain element in the interpolation space

Interpolation in infinite dimensions 1/2

- Use as an index-set if working on $\mathbb{R}^{\mathbb{N}}$:

$$\mathscr{F} := \{\nu \in \ell^\infty(\mathbb{N}) : \|\nu\|_0 < \infty\}, \quad \|\nu\|_0 = |\{j \in \mathbb{N} : \ \nu_j \neq 0\}|$$

- An interpolation operator is well-defined if it uniquely defines a certain element in the interpolation space

Definition

Let $\Lambda \subset \mathscr{F}$ be a set with cardinality $|\Lambda| = N$. A set $\Gamma \subseteq [-1, 1]^{\mathbb{N}}$ with cardinality $|\Gamma| = N$ is called *unisolvent* for $\mathcal{P}_\Lambda$ if any element in $\mathcal{P}_\Lambda$ is uniquely determined by its values on Γ.

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, ...) \in \mathscr{F}$ define:

- Interpolation

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, \ldots) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j\geq 1} \Gamma_{\nu_j}$

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, \ldots) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j \geq 1} \Gamma_{\nu_j}$
  - Operator: $\mathcal{I}_\nu := \bigotimes_{j \geq 1} \mathcal{I}_{\nu_j}$

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, ...) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j\geq 1} \Gamma_{\nu_j}$
  - Operator: $\mathcal{I}_\nu := \bigotimes_{j\geq 1} \mathcal{I}_{\nu_j}$
  - Space: $\mathcal{P}_\nu := \bigotimes_{j\geq 1} \mathcal{P}_{\nu_j}$

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, ...) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j \geq 1} \Gamma_{\nu_j}$
  - Operator: $\mathcal{I}_\nu := \bigotimes_{j \geq 1} \mathcal{I}_{\nu_j}$
  - Space: $\mathcal{P}_\nu := \bigotimes_{j \geq 1} \mathcal{P}_{\nu_j}$
- Is that well-defined (i.e. unisolvent)?

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, ...) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j \geq 1} \Gamma_{\nu_j}$
  - Operator: $\mathcal{I}_\nu := \bigotimes_{j \geq 1} \mathcal{I}_{\nu_j}$
  - Space: $\mathcal{P}_\nu := \bigotimes_{j \geq 1} \mathcal{P}_{\nu_j}$
- Is that well-defined (i.e. unisolvent)?
- Difference operator: $\Delta_\nu := \bigotimes_{j \geq 1} \Delta_{\nu_j}$

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, \dots) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j \geq 1} \Gamma_{\nu_j}$
  - Operator: $\mathcal{I}_\nu := \bigotimes_{j \geq 1} \mathcal{I}_{\nu_j}$
  - Space: $\mathcal{P}_\nu := \bigotimes_{j \geq 1} \mathcal{P}_{\nu_j}$
- Is that well-defined (i.e. unisolvent)?
- Difference operator: $\Delta_\nu := \bigotimes_{j \geq 1} \Delta_{\nu_j}$
- Hierarchical basis: $\{H_{\tilde{\nu}} : \tilde{\nu} \leq \nu\}$, $H_\nu(y) := \prod_{\nu_j \neq 0} h_{\nu_j}(y_j)$

Interpolation in infinite dimensions 2/2

For any index $\nu = (\nu_1, \ldots) \in \mathscr{F}$ define:

- Interpolation
  - Grid: $\Gamma_\nu := \bigotimes_{j\geq 1} \Gamma_{\nu_j}$
  - Operator: $\mathcal{I}_\nu := \bigotimes_{j\geq 1} \mathcal{I}_{\nu_j}$
  - Space: $\mathcal{P}_\nu := \bigotimes_{j\geq 1} \mathcal{P}_{\nu_j}$
- Is that well-defined (i.e. unisolvent)?
- Difference operator: $\Delta_\nu := \bigotimes_{j\geq 1} \Delta_{\nu_j}$
- Hierarchical basis: $\{H_{\tilde{\nu}} : \ \tilde{\nu} \leq \nu\}$, $H_\nu(y) := \prod_{\nu_j \neq 0} h_{\nu_j}(y_j)$

Curse of dimensionality: $\dim \mathcal{P}_\nu = \prod_{j\geq 1}(1 + \nu_j)$

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.
- Without additional structure on Λ nothing is clear

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.
- Without additional structure on Λ nothing is clear
  - What is $\mathcal{P}_\Lambda$?

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.
- Without additional structure on Λ nothing is clear
  - What is $\mathcal{P}_\Lambda$?
  - Is Γ_Λ unisolvent?

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.
- Without additional structure on Λ nothing is clear
  - What is $\mathcal{P}_\Lambda$?
  - Is Γ_Λ unisolvent?

Solution:

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.
- Without additional structure on Λ nothing is clear
  - What is $\mathcal{P}_\Lambda$?
  - Is Γ_Λ unisolvent?

Solution:

- If $\nu \in \Lambda$, then $\tilde{\nu} \leq \nu$ should be in Λ

Sparse interpolation 1/3

- $\Lambda \subseteq \mathscr{F}$ **any** finite set
- Define the interpolation grid $\Gamma_\Lambda := \{y_\nu := (t_{\nu_j})_{j\geq 1},\ \nu \in \Lambda\}$
- Define the interpolation operator $\mathcal{I}_\Lambda := \sum_{\nu\in\Lambda} \Delta_\nu$.
- Without additional structure on Λ nothing is clear
  - What is $\mathcal{P}_\Lambda$?
  - Is Γ_Λ unisolvent?

Solution:

- If $\nu \in \Lambda$, then $\tilde{\nu} \leq \nu$ should be in Λ
- Demand Λ to be "downward closed"

Sparse interpolation 2/3

Definition

A generic index set $\Lambda \subseteq \mathscr{F}$ is *downward closed* if for any $\nu \in \Lambda$, the property $\tilde{\nu} \leq \nu$ implies $\tilde{\nu} \in \Lambda$.

Sparse interpolation 2/3

Definition

A generic index set $\Lambda \subseteq \mathscr{F}$ is *downward closed* if for any $\nu \in \Lambda$, the property $\tilde{\nu} \leq \nu$ implies $\tilde{\nu} \in \Lambda$.

Theorem

Let $\Lambda \subseteq \mathscr{F}$ be a finite downward closed set. Then, Γ_Λ is unisolvent for

$$\mathcal{P}_\Lambda := \mathrm{span}\{y \mapsto y^\nu,\ \nu \in \Lambda\}, \quad y^\nu := \prod_{\nu_j \neq 0} y_j^{\nu_j},$$

and $\mathcal{I}_\Lambda$ is the corresponding interpolation operator.

Sparse interpolation 3/3

Conclude:

Sparse interpolation 3/3

Conclude:

$\mathcal{I}_\Lambda = \sum_{\nu \in \Lambda} \Delta_\nu$ is well-defined as interpolation on $\mathcal{P}_\Lambda$

Sparse interpolation 3/3

Conclude:

$\mathcal{I}_\Lambda = \sum_{\nu \in \Lambda} \Delta_\nu$ is well-defined as interpolation on $\mathcal{P}_\Lambda$

What we do not know (yet):

Sparse interpolation 3/3

Conclude:

$\mathcal{I}_\Lambda = \sum_{\nu \in \Lambda} \Delta_\nu$ is well-defined as interpolation on $\mathcal{P}_\Lambda$

What we do not know (yet):

- How can we compute $\mathcal{I}_\Lambda u$ for the solution u?

Sparse interpolation 3/3

Conclude:

$\mathcal{I}_\Lambda = \sum_{\nu\in\Lambda} \Delta_\nu$ is well-defined as interpolation on $\mathcal{P}_\Lambda$

What we do not know (yet):

- How can we compute $\mathcal{I}_\Lambda u$ for the solution u?
- How well does it approximate in the infinite-dim. $[-1,1]^{\mathbb{N}}$?

Computing sparse interpolation 1/2

- $\Lambda := \Lambda_N$ has a partial ordering "$\leq$": $\Lambda_N = \{\nu_1, ..., \nu_N\}$

Computing sparse interpolation 1/2

- $\Lambda := \Lambda_N$ has a partial ordering "$\leq$": $\Lambda_N = \{\nu_1, ..., \nu_N\}$
- Remove a maximal element $\nu_N \to \Lambda_{N-1}$ downward closed

Computing sparse interpolation 1/2

- $\Lambda := \Lambda_N$ has a partial ordering "$\leq$": $\Lambda_N = \{\nu_1, ..., \nu_N\}$
- Remove a maximal element $\nu_N \to \Lambda_{N-1}$ downward closed
- Write difference operator as

$$\Delta_{\nu_n} u = \alpha_{\nu_n} H_{\nu_n}, \quad \alpha_{\nu_n} := u(y_{\nu_n}) - \mathcal{I}_{\Lambda_{n-1}} u(y_{\nu_n})$$

Computing sparse interpolation 1/2

- $\Lambda := \Lambda_N$ has a partial ordering "$\leq$": $\Lambda_N = \{\nu_1, ..., \nu_N\}$
- Remove a maximal element $\nu_N \to \Lambda_{N-1}$ downward closed
- Write difference operator as

$$\Delta_{\nu_n} u = \alpha_{\nu_n} H_{\nu_n}, \quad \alpha_{\nu_n} := u(y_{\nu_n}) - \mathcal{I}_{\Lambda_{n-1}} u(y_{\nu_n})$$

- Compute recursively:

$$\mathcal{I}_{\Lambda_n} u = \mathcal{I}_{\Lambda_{n-1}} u + \Delta_{\nu_n} u = \mathcal{I}_{\Lambda_{n-1}} u + \alpha_{\nu_n} H_{\nu_n}, \quad \mathcal{I}_{\Lambda_0} \equiv 0$$

Computing sparse interpolation 2/2

Algorithm:

Computing sparse interpolation 2/2

Algorithm:

Until error small

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

How do we pick the next multi-index?

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

How do we pick the next multi-index?

- non-adaptively: a-priori choice of nested sequences $(\Lambda_n)_n$

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

How do we pick the next multi-index?

- non-adaptively: a-priori choice of nested sequences $(\Lambda_n)_n$
- adaptively: **Try** to find the N largest $\|\Delta_{\nu_n} u\|_{L^p}$

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

How do we pick the next multi-index?

- non-adaptively: a-priori choice of nested sequences $(\Lambda_n)_n$
- adaptively: **Try** to find the N largest $\|\Delta_{\nu_n} u\|_{L^p}$
  - Find points which maximize interpolation error

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

How do we pick the next multi-index?

- non-adaptively: a-priori choice of nested sequences $(\Lambda_n)_n$
- adaptively: **Try** to find the N largest $\|\Delta_{\nu_n} u\|_{L^p}$
  - Find points which maximize interpolation error
  - Attention to downward-closedness and other issues

Computing sparse interpolation 2/2

Algorithm:

Until error small, pick ν^{N+1}, compute $\mathcal{I}_{\Lambda_{N+1}}$ recursively

How do we pick the next multi-index?

- non-adaptively: a-priori choice of nested sequences $(\Lambda_n)_n$
- adaptively: **Try** to find the N largest $\|\Delta_{\nu_n} u\|_{L^p}$
  - Find points which maximize interpolation error
  - Attention to downward-closedness and other issues
  - In practice good behaviour; proof of convergence is **open**

Convergence and stability of sparse interpolation

Convergence is stability

- $\mathcal{I}_\Lambda$ is projection from $C([-1,1];V)$ onto $\operatorname{ran}\mathcal{I}_\Lambda$

Convergence is stability

- $\mathcal{I}_\Lambda$ is projection from $C([-1,1];V)$ onto $\operatorname{ran}\mathcal{I}_\Lambda$
- Interpolation stable $\leftrightarrow$ $\mathbb{L}_\Lambda = \sup \frac{\|\mathcal{I}_\Lambda u\|_{L^\infty}}{\|u\|_{L^\infty}}$ bounded

Convergence is stability

- $\mathcal{I}_\Lambda$ is projection from $C([-1,1];V)$ onto $\operatorname{ran}\mathcal{I}_\Lambda$
- Interpolation stable $\leftrightarrow$ $\mathbb{L}_\Lambda = \sup \frac{\|\mathcal{I}_\Lambda u\|_{L^\infty}}{\|u\|_{L^\infty}}$ bounded

Error estimate needs stability:

Convergence is stability

- $\mathcal{I}_\Lambda$ is projection from $C([-1,1];V)$ onto $\operatorname{ran}\mathcal{I}_\Lambda$
- Interpolation stable $\leftrightarrow$ $\mathbb{L}_\Lambda = \sup \frac{\|\mathcal{I}_\Lambda u\|_{L^\infty}}{\|u\|_{L^\infty}}$ bounded

Error estimate needs stability:

- Use $v = \mathcal{I}_\Lambda v$ for $v \in \operatorname{ran}\mathcal{I}_\Lambda$ and operator norm of $\mathcal{I}_\Lambda$:

Convergence is stability

- $\mathcal{I}_\Lambda$ is projection from $C([-1,1];V)$ onto $\operatorname{ran}\mathcal{I}_\Lambda$
- Interpolation stable $\leftrightarrow$ $\mathbb{L}_\Lambda = \sup \frac{\|\mathcal{I}_\Lambda u\|_{L^\infty}}{\|u\|_{L^\infty}}$ bounded

Error estimate needs stability:

- Use $v = \mathcal{I}_\Lambda v$ for $v \in \operatorname{ran}\mathcal{I}_\Lambda$ and operator norm of $\mathcal{I}_\Lambda$:

$$\begin{aligned} \|u - \mathcal{I}_\Lambda u\|_{L^\infty} &\leq \|u - v_\Lambda\|_{L^\infty} + \|\mathcal{I}_\Lambda(u - v_\Lambda)\|_{L^\infty} \\ &\leq (1 + \mathbb{L}_\Lambda) \inf_{v_\Lambda \in \operatorname{ran}\mathcal{I}_\Lambda} \|u - v_\Lambda\|_{L^\infty}. \end{aligned}$$

Convergence is stability

- $\mathcal{I}_\Lambda$ is projection from $C([-1,1];V)$ onto $\operatorname{ran}\mathcal{I}_\Lambda$
- Interpolation stable $\leftrightarrow$ $\mathbb{L}_\Lambda = \sup \frac{\|\mathcal{I}_\Lambda u\|_{L^\infty}}{\|u\|_{L^\infty}}$ bounded

Error estimate needs stability:

- Use $v = \mathcal{I}_\Lambda v$ for $v \in \operatorname{ran}\mathcal{I}_\Lambda$ and operator norm of $\mathcal{I}_\Lambda$:

$$
\begin{aligned}
\|u - \mathcal{I}_\Lambda u\|_{L^\infty} &\leq \|u - v_\Lambda\|_{L^\infty} + \|\mathcal{I}_\Lambda(u - v_\Lambda)\|_{L^\infty} \\
&\leq (1 + \mathbb{L}_\Lambda) \inf_{v_\Lambda \in \operatorname{ran}\mathcal{I}_\Lambda} \|u - v_\Lambda\|_{L^\infty}.
\end{aligned}
$$

- Hopefully $\mathbb{L}_\Lambda$ smaller than bound for $\inf_{v_\Lambda \in \operatorname{ran}\mathcal{I}_\Lambda} \|u - v_\Lambda\|$

Best N-term approximation

- (Anisotropic) smoothness of $y \mapsto u(y)$

Best N-term approximation

- (Anisotropic) smoothness of $y \mapsto u(y)$
- Use Taylor expansion: $u(y) = \sum_{\nu \in \mathscr{F}} t_\nu T_\nu$,

Best N-term approximation

- (Anisotropic) smoothness of $y \mapsto u(y)$
- Use Taylor expansion: $u(y) = \sum_{\nu \in \mathscr{F}} t_\nu T_\nu$,
- Purely functional analytic statement:

Best N-term approximation

- (Anisotropic) smoothness of $y \mapsto u(y)$
- Use Taylor expansion: $u(y) = \sum_{\nu \in \mathscr{F}} t_\nu T_\nu$,
- Purely functional analytic statement:

$$\inf_{v \in \operatorname{ran} \mathcal{I}_\Lambda} \|u - v\|_{L^\infty} \leq \left\| \sum_{\nu \notin \Lambda} t_\nu T_\nu \right\|_{L^\infty} \leq \sum_{\nu \notin \Lambda} \|t_\nu\|_V$$

Best N-term approximation

- (Anisotropic) smoothness of $y \mapsto u(y)$
- Use Taylor expansion: $u(y) = \sum_{\nu \in \mathscr{F}} t_\nu T_\nu$,
- Purely functional analytic statement:

$$\inf_{v \in \operatorname{ran} \mathcal{I}_\Lambda} \|u - v\|_{L^\infty} \leq \left\| \sum_{\nu \notin \Lambda} t_\nu T_\nu \right\|_{L^\infty} \leq \sum_{\nu \notin \Lambda} \|t_\nu\|_V$$

- "Stechkin"

Best N-term approximation

- (Anisotropic) smoothness of $y \mapsto u(y)$
- Use Taylor expansion: $u(y) = \sum_{\nu \in \mathscr{F}} t_\nu T_\nu$,
- Purely functional analytic statement:

$$\inf_{v \in \operatorname{ran} \mathcal{I}_\Lambda} \|u - v\|_{L^\infty} \leq \left\| \sum_{\nu \notin \Lambda} t_\nu T_\nu \right\|_{L^\infty} \leq \sum_{\nu \notin \Lambda} \|t_\nu\|_V$$

- "Stechkin": if $(\|t_\nu\|_\nu)_{\nu \in \mathscr{F}} \in \ell^p(\mathbb{N}_0)$ for $p < 1$:

$$\sum_{\nu \notin \Lambda} \|t_\nu\|_V \leq N^{-s}, \quad s = \frac{1}{p} - 1, \quad N = |\Lambda|$$

Bounding the Lebesgue constant

Introduce **univariate** Lebesgue constants for $k \geq 0$:

$$\lambda_k := \sup \frac{\|\mathcal{I}_k u\|_{L^\infty}}{\|u\|_{L^\infty}}, \quad \delta_k := \sup \frac{\|\Delta_k u\|_{L^\infty}}{\|u\|_{L^\infty}}$$

Bounding the Lebesgue constant

Introduce **univariate** Lebesgue constants for $k \geq 0$:

$$\lambda_k := \sup \frac{\|\mathcal{I}_k u\|_{L^\infty}}{\|u\|_{L^\infty}}, \quad \delta_k := \sup \frac{\|\Delta_k u\|_{L^\infty}}{\|u\|_{L^\infty}}$$

Theorem

If there exists $\theta \geq 1$ such that $\lambda_k \leq (k+1)^\theta$ or $\delta_k \leq (k+1)^\theta$ holds for $k \geq 0$, then the Lebesgue constant $\mathbb{L}_\Lambda$ satisfies $\mathbb{L}_\Lambda \leq N^{\theta+1}$ for any downward closed set Λ with $|\Lambda| = N$.

Bounding the Lebesgue constant

Introduce **univariate** Lebesgue constants for $k \geq 0$:

$$\lambda_k := \sup \frac{\|\mathcal{I}_k u\|_{L^\infty}}{\|u\|_{L^\infty}}, \quad \delta_k := \sup \frac{\|\Delta_k u\|_{L^\infty}}{\|u\|_{L^\infty}}$$

Theorem

If there exists $\theta \geq 1$ such that $\lambda_k \leq (k+1)^\theta$ or $\delta_k \leq (k+1)^\theta$ holds for $k \geq 0$, then the Lebesgue constant $\mathbb{L}_\Lambda$ satisfies $\mathbb{L}_\Lambda \leq N^{\theta+1}$ for any downward closed set Λ with $|\Lambda| = N$.

Sketch of the proof.

Derive bounds for $\mathbb{L}_\Lambda$ using a collection of $\delta_\cdot$; then one case is clear, the other one will be clear inductively. $\square$

Which bounds can we obtain?

Conclude from the theorem:

$\mathbb{L}_\Lambda$ "inherits" bounds from univariate Lebesgue constants

Which bounds can we obtain?

Conclude from the theorem:

$\mathbb{L}_\Lambda$ "inherits" bounds from univariate Lebesgue constants

- Clenshaw-Curtis points: $\lambda_k \sim \log(k)$ but not nested

Which bounds can we obtain?

Conclude from the theorem:

$\mathbb{L}_\Lambda$ "inherits" bounds from univariate Lebesgue constants

- Clenshaw-Curtis points: $\lambda_k \sim \log(k)$ but not nested
- Leja points: $t_0 = 0$ and t_k given by

$$t_k := \arg\max \left\{ \prod_{i \neq j} |t_i - t_j| : \ \{t_0, ..., t_k\} \in [-1, 1]^{k+1} \right\}$$

Which bounds can we obtain?

Conclude from the theorem:

$\mathbb{L}_\Lambda$ "inherits" bounds from univariate Lebesgue constants

- Clenshaw-Curtis points: $\lambda_k \sim \log(k)$ but not nested
- Leja points: $t_0 = 0$ and t_k given by

$$t_k := \arg\max \left\{ \prod_{i \neq j} |t_i - t_j| : \ \{t_0, \dots, t_k\} \in [-1, 1]^{k+1} \right\}$$

- Bound is **promising** to be $\lambda_k \leq (1 + k)$ - open problem!

Which bounds can we obtain?

Conclude from the theorem:

$\mathbb{L}_\Lambda$ "inherits" bounds from univariate Lebesgue constants

- Clenshaw-Curtis points: $\lambda_k \sim \log(k)$ but not nested
- Leja points: $t_0 = 0$ and t_k given by

$$t_k := \arg\max \left\{ \prod_{i \neq j} |t_i - t_j| : \ \{t_0, ..., t_k\} \in [-1,1]^{k+1} \right\}$$

- Bound is **promising** to be $\lambda_k \leq (1+k)$ - open problem!
- So-called $\mathfrak{R}$-Leja points: $\lambda_k \leq (1+k)^2 \Rightarrow \mathbb{L}_\Lambda \leq N^3$

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$
- Bestapproximation in $\operatorname{ran}\mathcal{I}_\Lambda$ gives error N^{-s}

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$
- Bestapproximation in $\operatorname{ran} \mathcal{I}_\Lambda$ gives error N^{-s}
- $\mathbb{L}_\Lambda$ inherits bound from λ_k or δ_k

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$
- Bestapproximation in $\operatorname{ran} \mathcal{I}_\Lambda$ gives error N^{-s}
- $\mathbb{L}_\Lambda$ inherits bound from λ_k or δ_k

Extensions:

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$
- Bestapproximation in $\operatorname{ran} \mathcal{I}_\Lambda$ gives error N^{-s}
- $\mathbb{L}_\Lambda$ inherits bound from λ_k or δ_k

Extensions:

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$
- Bestapproximation in $\operatorname{ran} \mathcal{I}_\Lambda$ gives error N^{-s}
- $\mathbb{L}_\Lambda$ inherits bound from λ_k or δ_k

Extensions:

- Sharper error-estimate: N^{-s} error instead of $N^{-(s-1-\theta)}$

Summary

- Interpolation on $[-1,1]^{\mathbb{N}}$ via Δ_k and downward closed sets
- Compute via recursion formula
- Convergence boils down to stability - bound $\mathbb{L}_\Lambda$
- Bestapproximation in $\operatorname{ran} \mathcal{I}_\Lambda$ gives error N^{-s}
- $\mathbb{L}_\Lambda$ inherits bound from λ_k or δ_k

Extensions:

- Sharper error-estimate: N^{-s} error instead of $N^{-(s-1-\theta)}$
- Non-polynomial interpolation: RKHS, piecewise linear functions ($\to$ sparse grids), ...

Conclusion

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν
3. They are well-defined on downward closed index sets

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν
3. They are well-defined on downward closed index sets
4. Convergence reduces to bounding $\mathbb{L}_\Lambda$

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν
3. They are well-defined on downward closed index sets
4. Convergence reduces to bounding $\mathbb{L}_\Lambda$
5. $\mathbb{L}_\Lambda$ can be bounded by $N^{\theta+1}$ or better

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν
3. They are well-defined on downward closed index sets
4. Convergence reduces to bounding $\mathbb{L}_\Lambda$
5. $\mathbb{L}_\Lambda$ can be bounded by $N^{\theta+1}$ or better
   if pointset is very nice $=$ (nested, evenly spaced)

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν
3. They are well-defined on downward closed index sets
4. Convergence reduces to bounding $\mathbb{L}_\Lambda$
5. $\mathbb{L}_\Lambda$ can be bounded by $N^{\theta+1}$ or better
   if pointset is very nice $=$ (nested, evenly spaced)
6. Best N-term approx. gives error $(1+\mathbb{L}_\Lambda)N^{-s}$ for $s = 1/p - 1$

Conclusion

1. Stochastic collocation is (high-dimensional) polynomial interpolation for vector-valued functions
2. Build sparse interpolation operators through Δ_ν
3. They are well-defined on downward closed index sets
4. Convergence reduces to bounding $\mathbb{L}_\Lambda$
5. $\mathbb{L}_\Lambda$ can be bounded by $N^{\theta+1}$ or better
   if pointset is very nice = (nested, evenly spaced)
6. Best N-term approx. gives error $(1+\mathbb{L}_\Lambda)N^{-s}$ for $s = 1/p - 1$
7. There is more to learn

The absolute minimum to remember

1. Stochastic collocation is interpolation

The absolute minimum to remember

1. Stochastic collocation is interpolation
2. Sparse interpolation: nested pointsets

The absolute minimum to remember

1. Stochastic collocation is interpolation
2. Sparse interpolation: nested pointsets
3. Well-defined on downward closed index sets

The absolute minimum to remember

1. Stochastic collocation is interpolation
2. Sparse interpolation: nested pointsets
3. Well-defined on downward closed index sets
4. Analysis: N-term approx. & Lebesgue const.

Main references

I. Babuška, F. Nobile, and R. Tempone.
A stochastic collocation method for elliptic partial differential equations with random input data.
SIAM Review, 52(2), 317-355. 2010.

A. Cohen and R. DeVore.
Approximation of high-dimensional parametric PDEs.
Acta Numerica, 24:1-154, 2015.
Sections 6.1 and 6.2.