IMSE Conference 2018

Tensor–product interpolation for high–dimensional integrals

Dmitry Savostyanov
University of Brighton, UK

Brighton 16/07/2018

High–dimensional integrals

- Generalised Gaussian integrals in mathematical finance

$$\int_{\mathbb{R}^d} f(x) e^{-\frac{1}{2}x^T A x} dx$$

- Ising integrals in mathematical physics

$$\int_{[0,1]^{d-1}} \left(\prod_{j\leqslant k} \frac{t_k t_{k-1} \cdots t_{j+1} - 1}{t_k t_{k-1} \cdots t_{j+1} + 1} \right)^2 dt$$

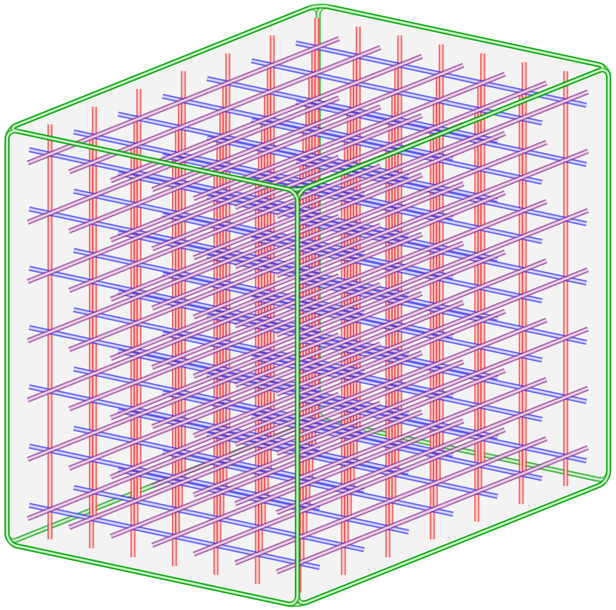
- Weyl's integrals in representation theory / random matrix theory

$$\int_{\mathbb{U}_d} f(U) dU = \frac{1}{(2\pi)^d d!} \int_{[0,2\pi]^d} f(\mathbf{diag}(e^{i\theta_1}, \ldots, e^{i\theta_d})) \prod_{j\leqslant k} \left| e^{i\theta_j} - e^{i\theta_k} \right|^2 d\theta$$

- Stochastic PDEs

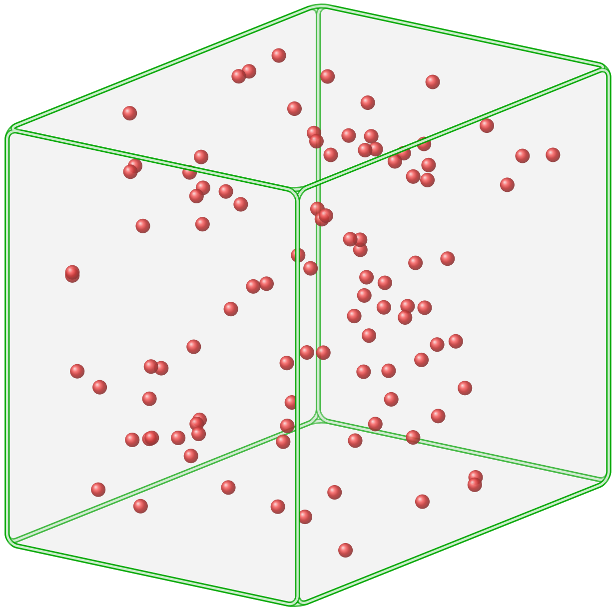
$$\mathcal{D}_x(\xi) u(x, \xi) = f(x)$$

Main message



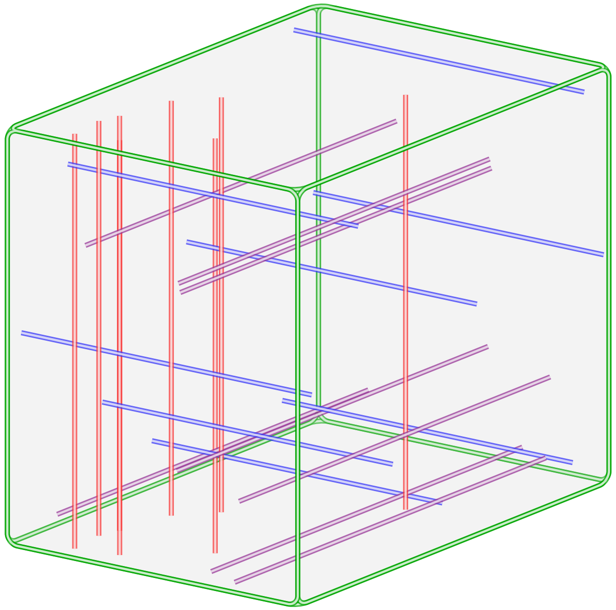
full grid

bad



Monte-Carlo

good



cross interpolation

better

Outline

- Introduction
- Cross interpolation theory
  - ⇒ Functions
  - — Matrices
  - — Tensors
- Cross interpolation practice

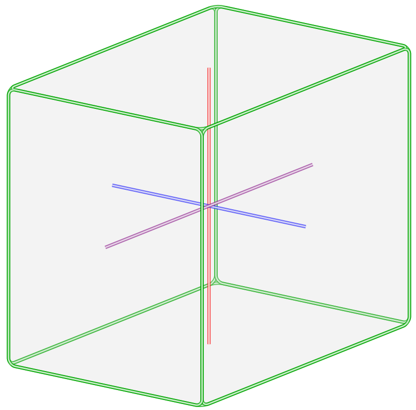
Cross interpolation (for functions)

$$\begin{aligned}\int_{\mathbb{R}^3} e^{-\frac{1}{2}\|x\|^2} dx &= \int_{\mathbb{R}}\int_{\mathbb{R}}\int_{\mathbb{R}} e^{-\frac{1}{2}(x_1^2+x_2^2+x_3^3)} dx_1 dx_2 dx_3 \\ &= \left(\int_{\mathbb{R}} e^{-\frac{1}{2}x_1^2} dx_1\right)\left(\int_{\mathbb{R}} e^{-\frac{1}{2}x_2^2} dx_2\right)\left(\int_{\mathbb{R}} e^{-\frac{1}{2}x_3^2} dx_3\right) \\ &= \left(\sqrt{2\pi}\right)^3\end{aligned}$$

Separation of variables

$$e^{-\frac{1}{2}(x_1^2+x_2^2+x_3^2)} = e^{-\frac{1}{2}x_1^2}\, e^{-\frac{1}{2}x_2^2}\, e^{-\frac{1}{2}x_3^2}$$

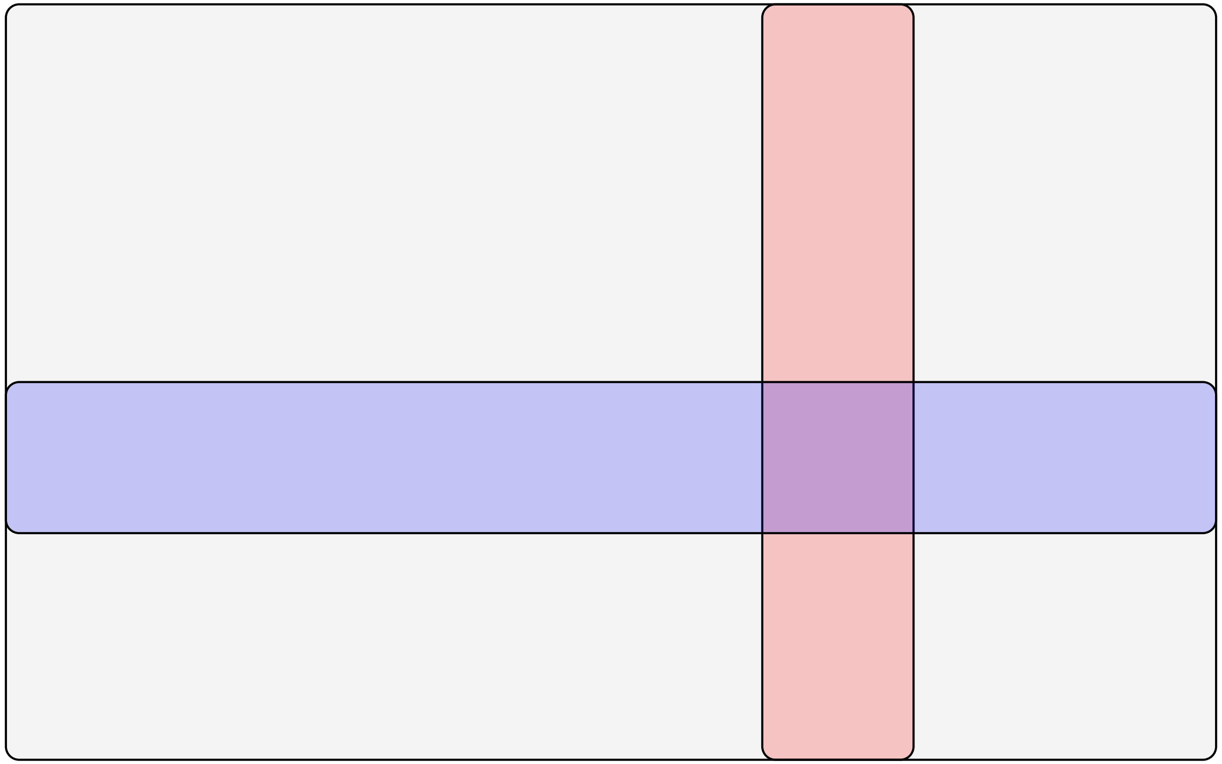
$$f(x_1, x_2, x_3) = f(x_1, 0, 0) f(0, x_2, 0) f(0, 0, x_3)$$



Outline

- Introduction
- Cross interpolation theory
  - Functions
  - ⇒ Matrices
  - Tensors
- Cross interpolation practice

Cross interpolation (for matrices)



$$A(i,j) \approx \tilde{A}(i,j) = A(i,\mathcal{J})\,[A(\mathcal{I},\mathcal{J})]^{-1}\,A(\mathcal{I},j).$$

$mr + rn - r^2$ parameters (SVD), $mr + rn - r^2$ interpolation points

Interpolation (good)

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \end{pmatrix}$$

$$= \begin{pmatrix} 1/2 & 1/3 \\ 1/3 & 1/4 \\ 1/4 & 1/5 \\ 1/5 & 1/6 \\ 1/6 & 1/7 \end{pmatrix} \cdot \begin{pmatrix} 1/2 & 1/3 \\ 1/3 & 1/4 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.67 & 2.86 & 3.57 \\ 0 & 0 & 2.86 & 5.00 & 6.35 \\ 0 & 0 & 3.57 & 6.35 & \mathbf{8.17} \end{pmatrix} \times \mathbf{10^{-3}}$$

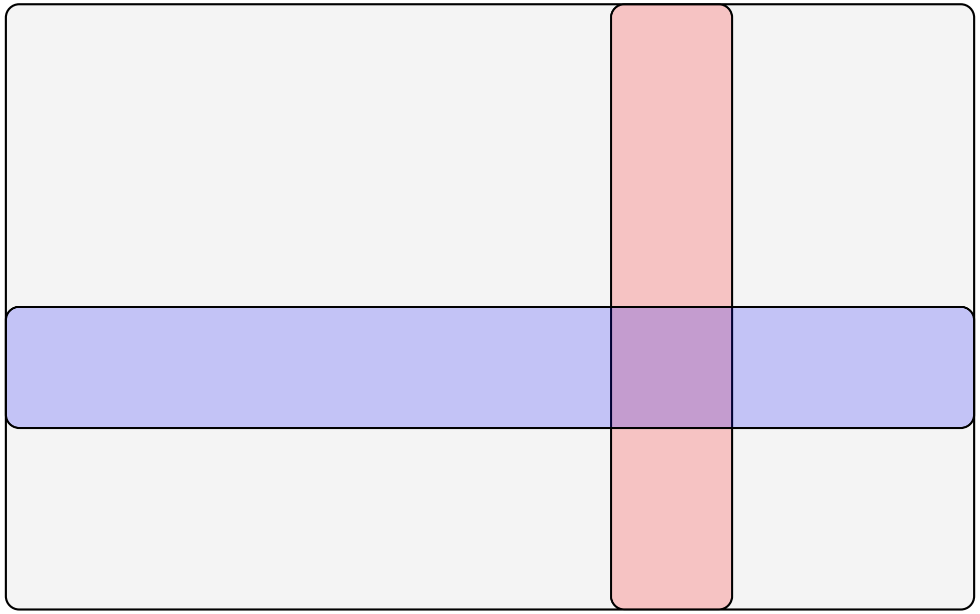
Interpolation (not so good)

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \end{pmatrix}$$

$$= \begin{pmatrix} 1/5 & 1/6 \\ 1/6 & 1/7 \\ 1/7 & 1/8 \\ 1/8 & 1/9 \\ 1/9 & 1/10 \end{pmatrix} \cdot \begin{pmatrix} 1/8 & 1/9 \\ 1/9 & 1/10 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \end{pmatrix}$$

$$+ \begin{pmatrix} \mathbf{8.00} & 1.90 & 0.36 & 0 & 0 \\ 1.90 & 0.51 & 0.10 & 0 & 0 \\ 0.36 & 0.10 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \times \mathbf{10^{-2}}$$

Quasioptimality [Goreinov, Tyrtyshnikov, 2001 & 2011]



$$[\mathcal{I},\mathcal{J}] = \mathsf{maxvol}[A(i,j)], \qquad \tilde{A}(i,j) = A(i,\mathcal{J})\,[A(\mathcal{I},\mathcal{J})]^{-1}\,A(\mathcal{I},j).$$

$$|A-\tilde{A}| \leqslant (r+1)\ \ \min_{\mathrm{rank}\,X=r} \|A-X\|$$

$$|A-\tilde{A}| \leqslant (r+1)^2 \min_{\mathrm{rank}\,X=r} |A-X|$$

Interpolation (maximum–volume)

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \end{pmatrix}$$

$$= \begin{pmatrix} 1/2 & 1/5 \\ 1/3 & 1/6 \\ 1/4 & 1/7 \\ 1/5 & 1/8 \\ 1/6 & 1/9 \end{pmatrix} \cdot \begin{pmatrix} 1/2 & 1/5 \\ 1/5 & 1/8 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{pmatrix}$$

$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{3.09} & 1.59 & 0 & -1.18 \\ 0 & 1.59 & 0.85 & 0 & -0.66 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1.18 & -0.66 & 0 & 0.55 \end{pmatrix} \times \mathbf{10^{-3}}$$

Interpolation (cross algorithm, step 1)

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \end{pmatrix}$$

$$= \begin{pmatrix} 1/2 \\ 1/3 \\ 1/4 \\ 1/5 \\ 1/6 \end{pmatrix} \cdot \begin{pmatrix} 1/2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \end{pmatrix}$$

$$- \quad \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2.78 & 3.33 & 3.33 & 3.17 \\ 0 & 3.33 & 4.17 & 4.28 & 4.17 \\ 0 & 3.33 & 4.28 & \mathbf{4.50} & 4.44 \\ 0 & 3.17 & 4.17 & 4.44 & 4.44 \end{pmatrix} \times 10^{-2}$$

Interpolation (cross algorithm, step 2)

$$A = \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & 1/7 & 1/8 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \\ 1/6 & 1/7 & 1/8 & 1/9 & 1/10 \end{pmatrix}$$

$$= \begin{pmatrix} 1/2 & 1/5 \\ 1/3 & 1/6 \\ 1/4 & 1/7 \\ 1/5 & 1/8 \\ 1/6 & 1/9 \end{pmatrix} \cdot \begin{pmatrix} 1/2 & 1/5 \\ 1/5 & 1/8 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/5 & 1/6 & 1/7 & 1/8 & 1/9 \end{pmatrix}$$

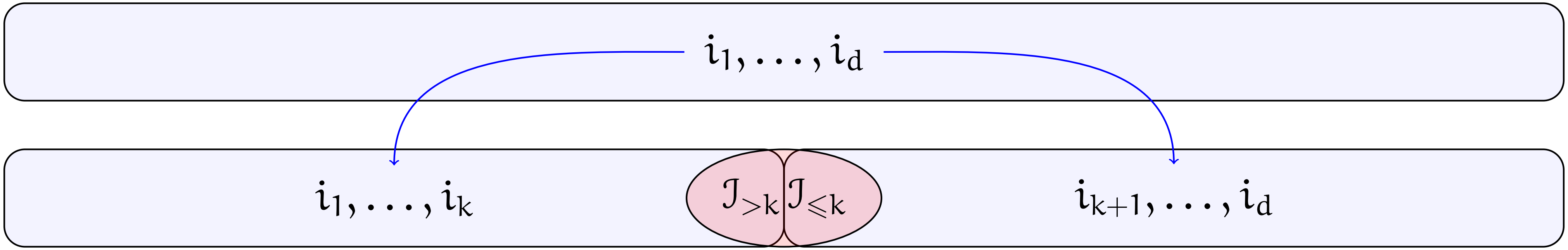
$$+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{3.09} & 1.59 & 0 & -1.18 \\ 0 & 1.59 & 0.85 & 0 & -0.66 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1.18 & -0.66 & 0 & 0.55 \end{pmatrix} \times \mathbf{10^{-3}}$$

Outline

- Introduction
- Cross interpolation theory
  - — Functions
  - — Matrices
  - ⇒ Tensors
- Cross interpolation practice

Matrix formula for tensor case

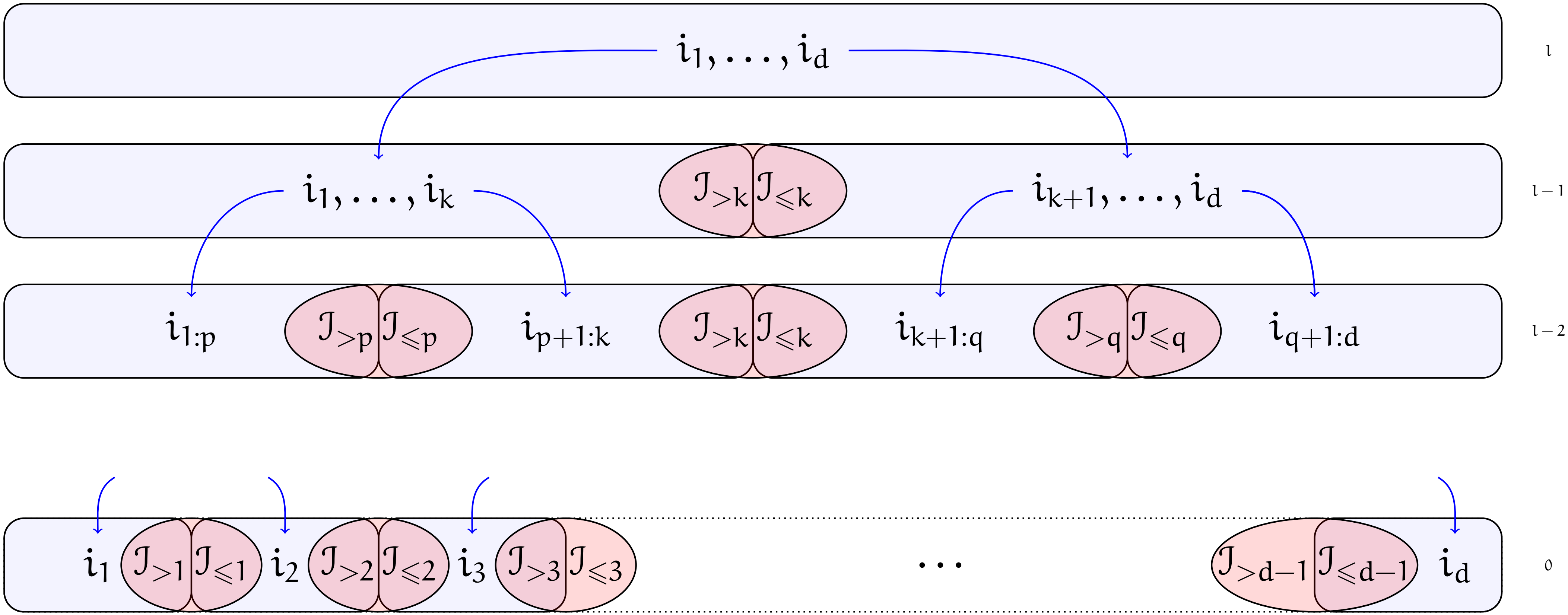
$$\begin{aligned} A(i_1,\ldots,i_k; i_{k+1},\ldots,i_d) &= A(i_{\leqslant k}, i_{>k}) \approx \tilde{A}(i_{\leqslant k}, i_{>k}) \\ &= A(i_{\leqslant k}, \mathcal{J}_{>k})\,[A(\mathcal{J}_{\leqslant k}, \mathcal{J}_{>k})]^{-1}\, A(\mathcal{J}_{\leqslant k}, i_{>k}) \\ &= A(i_1\ldots i_k, \mathcal{J}_{>k})\,[A(\mathcal{J}_{\leqslant k}, \mathcal{J}_{>k})]^{-1}\, A(\mathcal{J}_{\leqslant k}, i_{k+1}\ldots i_d) \end{aligned}$$



Error estimate

$$|A - \tilde{A}| \leqslant (r_k + 1)^2 \min |A - X|$$

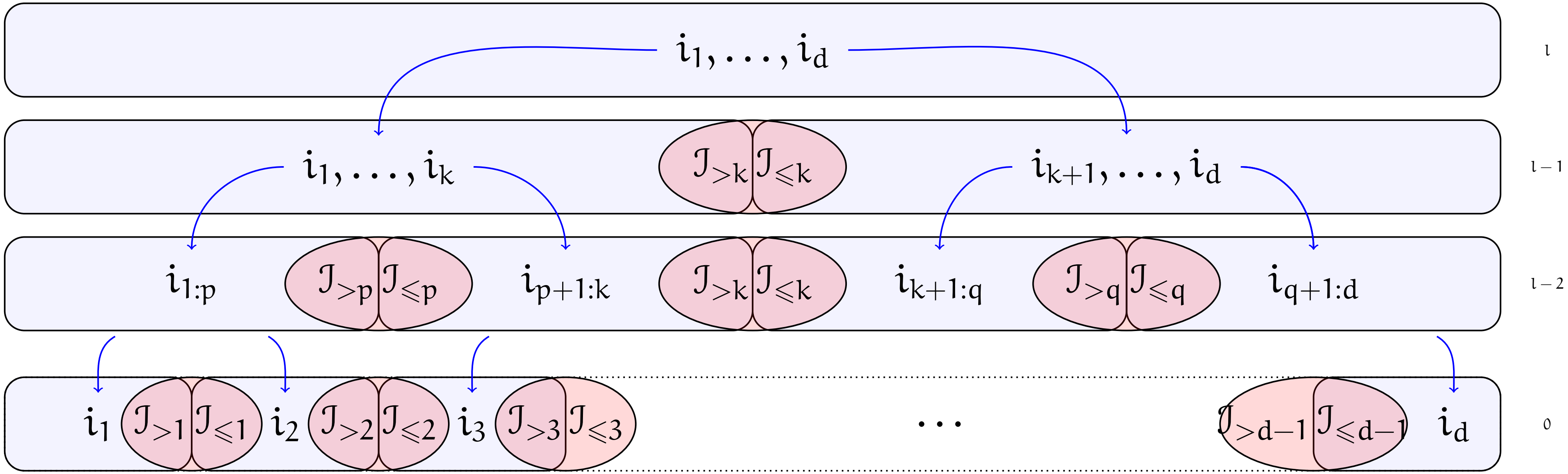
Separation of variables on a dimension tree



TT–cross interpolation formula [Oseledets, Tyrtyshnikov, 2010]

$$\tilde{A}(i_1, i_2 \ldots, i_d) = A(i_1, \mathcal{J}_{>1}) \, [A(\mathcal{J}_{\leqslant 1}, \mathcal{J}_{>1})]^{-1} \, A(\mathcal{J}_{\leqslant 1}, i_2, \mathcal{J}_{>2}) \, [A(\mathcal{J}_{\leqslant 2}, \mathcal{J}_{>2})]^{-1} \ldots A(\mathcal{J}_{\leqslant d-1}, i_d)$$

Theorem (Quasioptimality) [Savostyanov, LAA 2014]

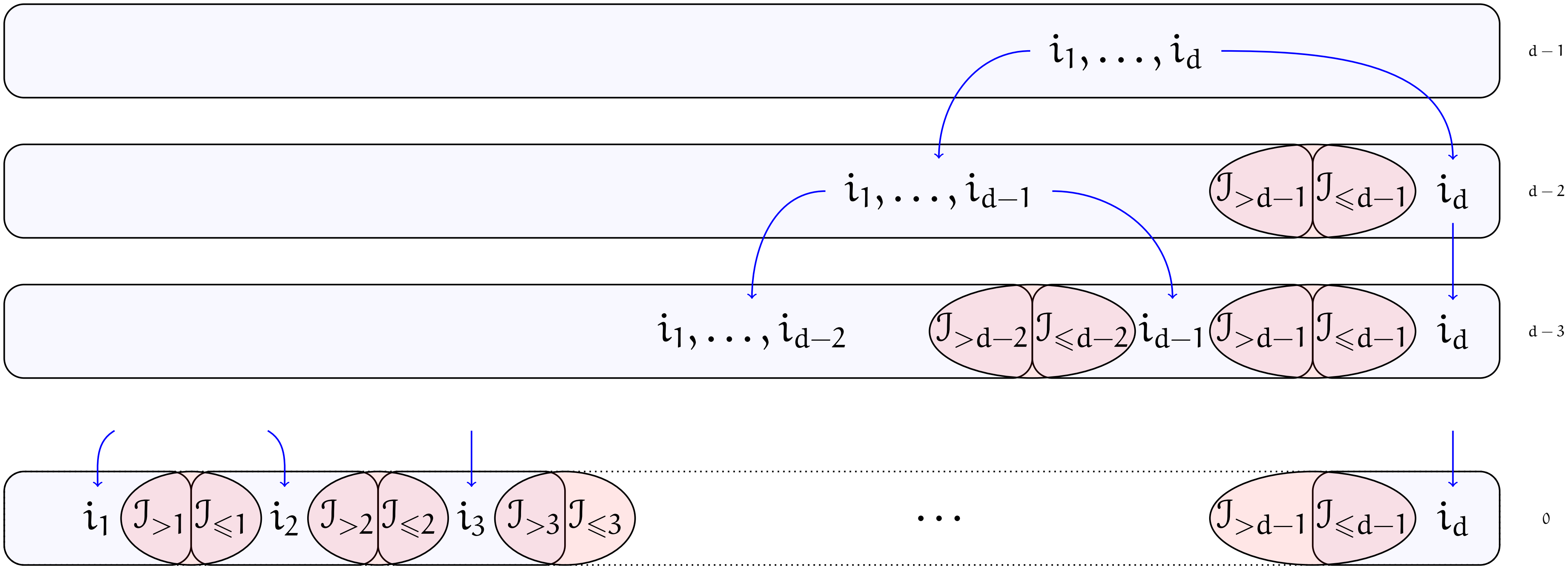


If $[\mathcal{J}_{\leqslant k}, \mathcal{J}_{>k}] = \text{maxvol}[A(i_{\leqslant k}, i_{>k})]$ then

$$|A - \tilde{A}| \leqslant (2r + \kappa r + 1)^{\lceil \log_2 d \rceil} (r+1)^2 |A - \tilde{A}_{\text{best}}|$$

$$r = \max r_k, \quad \kappa = \max \kappa_k, \quad \kappa_k = r_k |A| |A_k^{-1}|$$

Theorem (Interpolation) [Savostyanov, LAA 2014]



If indices are nested

$$i_{>k} \in \mathcal{J}_{>k} \;\Rightarrow\; i_{>k+1} \in \mathcal{J}_{>k+1}, \qquad i_{\leqslant k} \in \mathcal{J}_{\leqslant k} \;\Rightarrow\; i_{\leqslant k-1} \in \mathcal{J}_{\leqslant k-1},$$

then interpolation formula interpolates in all points

$$A(\mathcal{J}_{\leqslant k-1}, i_k, \mathcal{J}_{>k}) = \tilde{A}(\mathcal{J}_{\leqslant k-1}, i_k, \mathcal{J}_{>k}), \qquad k = 1, \ldots, d.$$

Outline

- Introduction
- Cross interpolation theory
- Cross interpolation practice

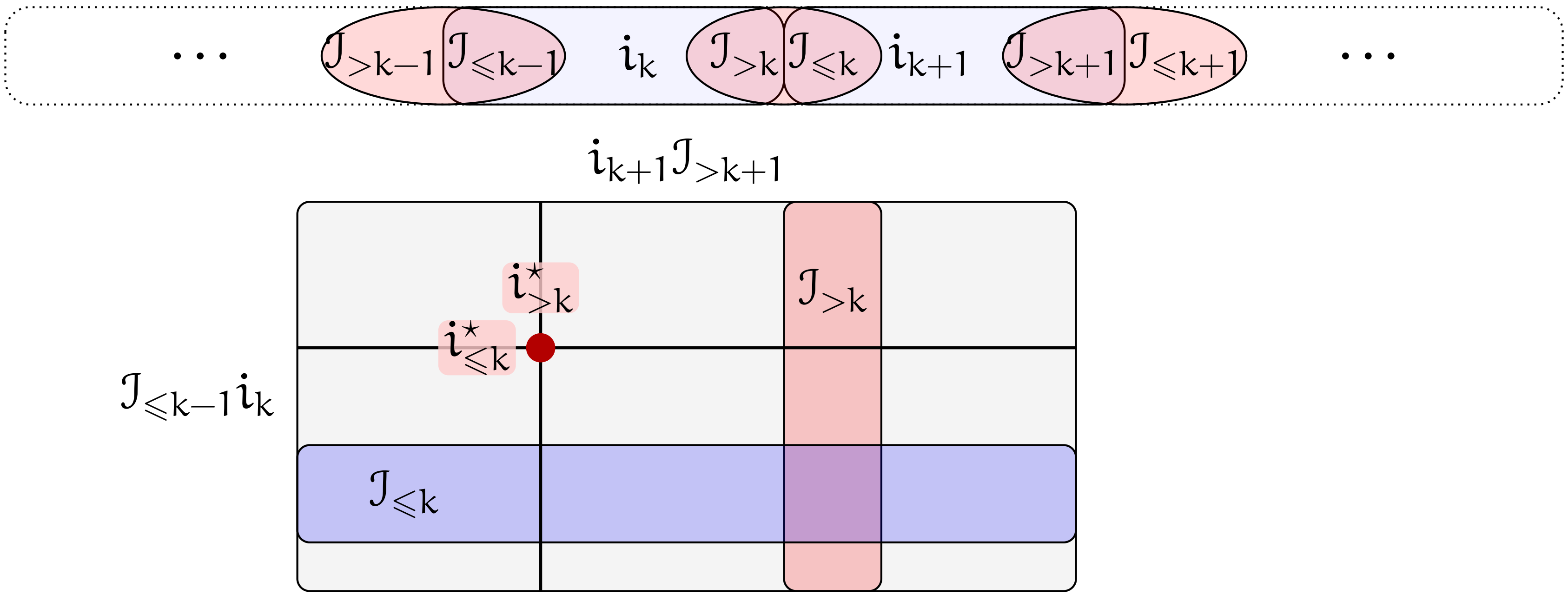
⇒ Cross interpolation algorithms for tensors

— High–dimensional integration

— Stochastic problems

— Parallel algorithm

DMRG–like restricted greedy interpolation [Savostyanov, LAA 2014]

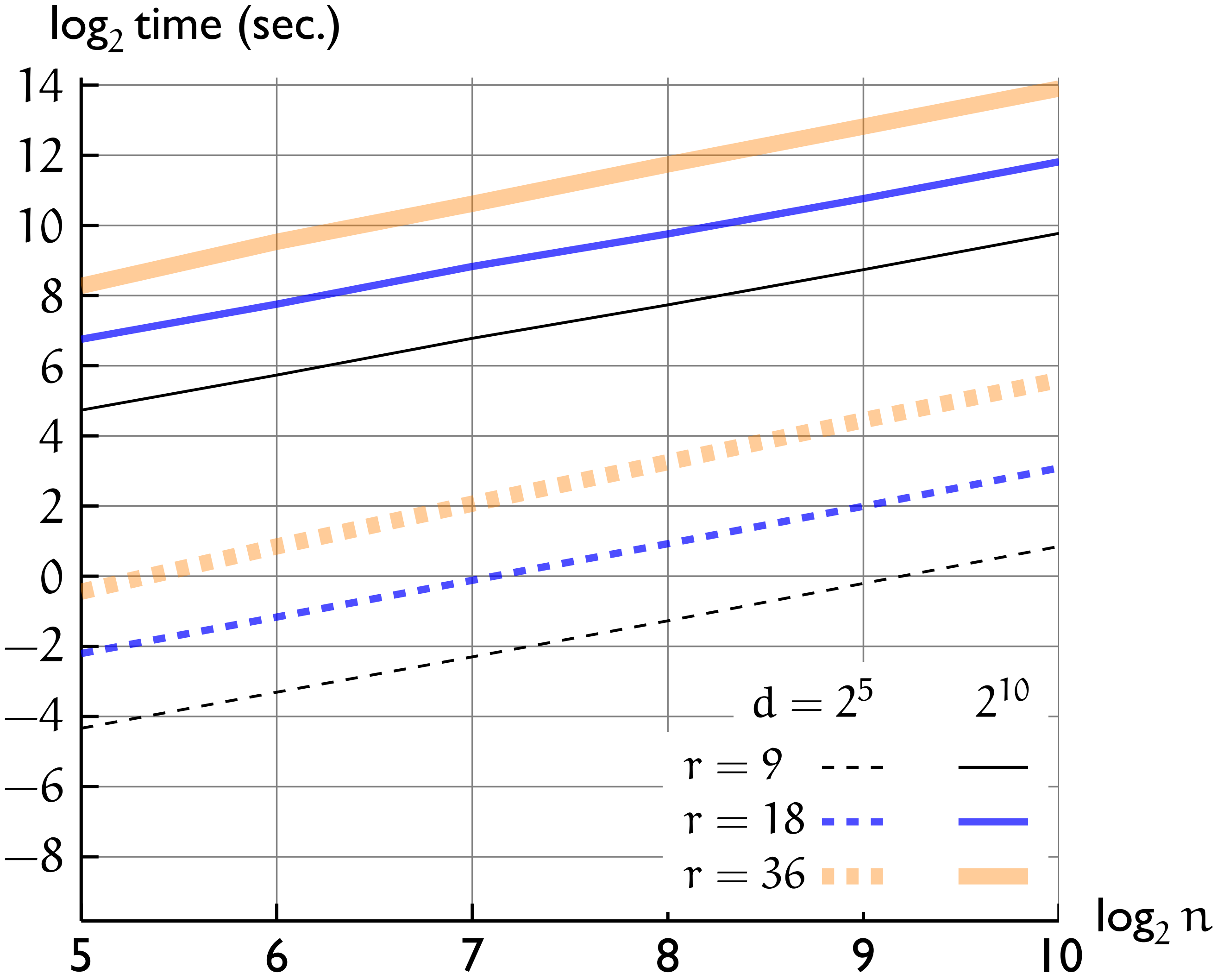


do $k = 1, \ldots, d-1$ (half–sweep)

- Find a pivot $i^\star = (i^\star_1, \ldots, i^\star_d)$ in $\left[A(\mathcal{J}^{\leqslant k-1} i_k, i_{k+1} \mathcal{J}^{>k+1})\right]$
- Add $i^\star_{\leqslant k}$ to $\mathcal{J}_{\leqslant k}$, and $i^\star_{>k}$ to $\mathcal{J}_{>k}$
- Update the TT–cores k and $k+1$ of the interpolation $\tilde{A}$

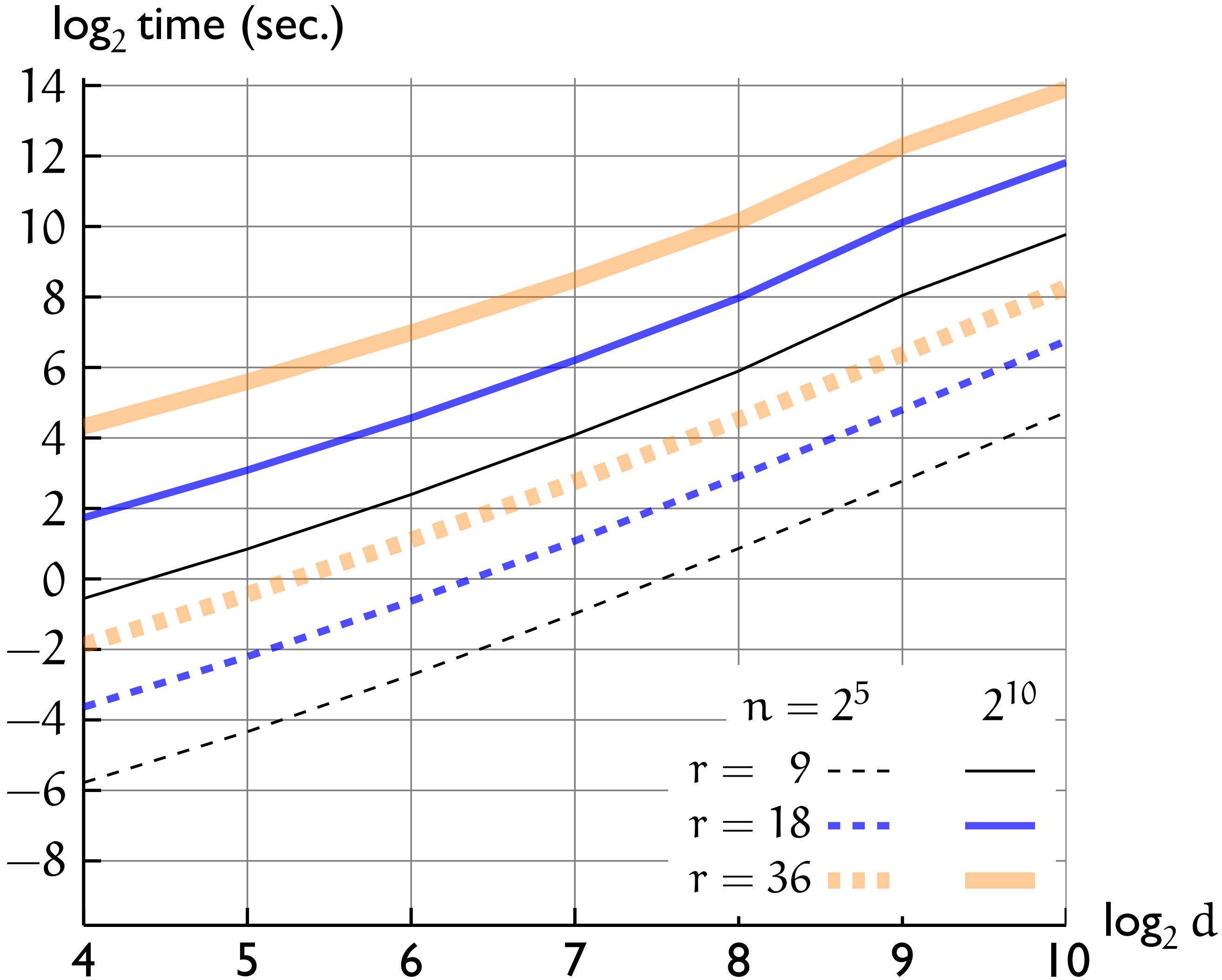
Complexity is $\sum_{k=1}^{d} r_{k-1} n_k r_k - \sum_{k=1}^{d-1} r_k$ tensor entries plus $\mathcal{O}(dnr^3)$ operations

Runtime wrt mode size for $A(i_1, \ldots, i_d) = 1/\sqrt{i_1^2 + \cdots + i_d^2}$



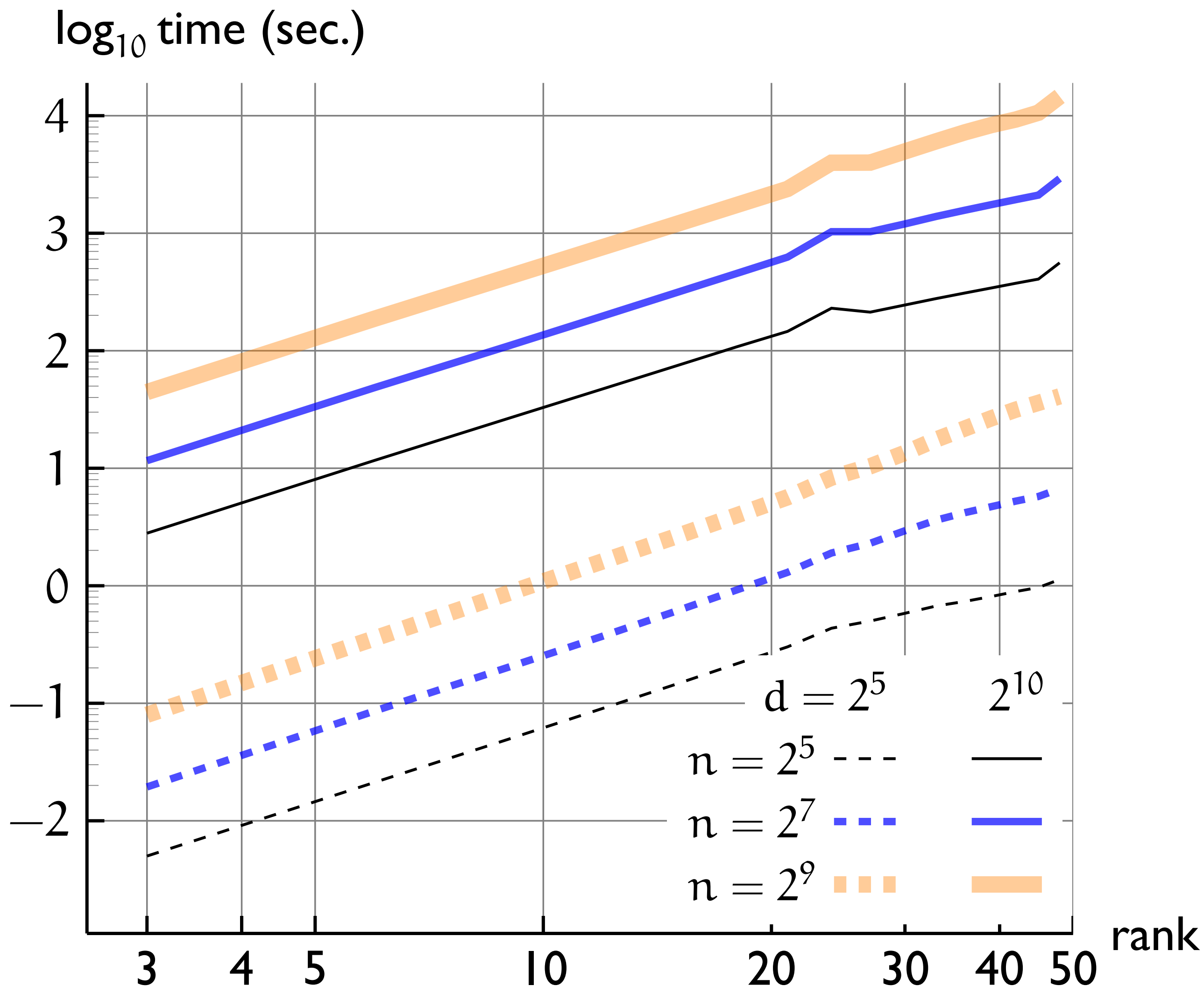
Strictly linear

Runtime wrt dimension for $A(i_1, \ldots, i_d) = 1/\sqrt{i_1^2 + \cdots + i_d^2}$



Scales from $\mathcal{O}(d^{1.5})$ to $\mathcal{O}(d^{1.8})$

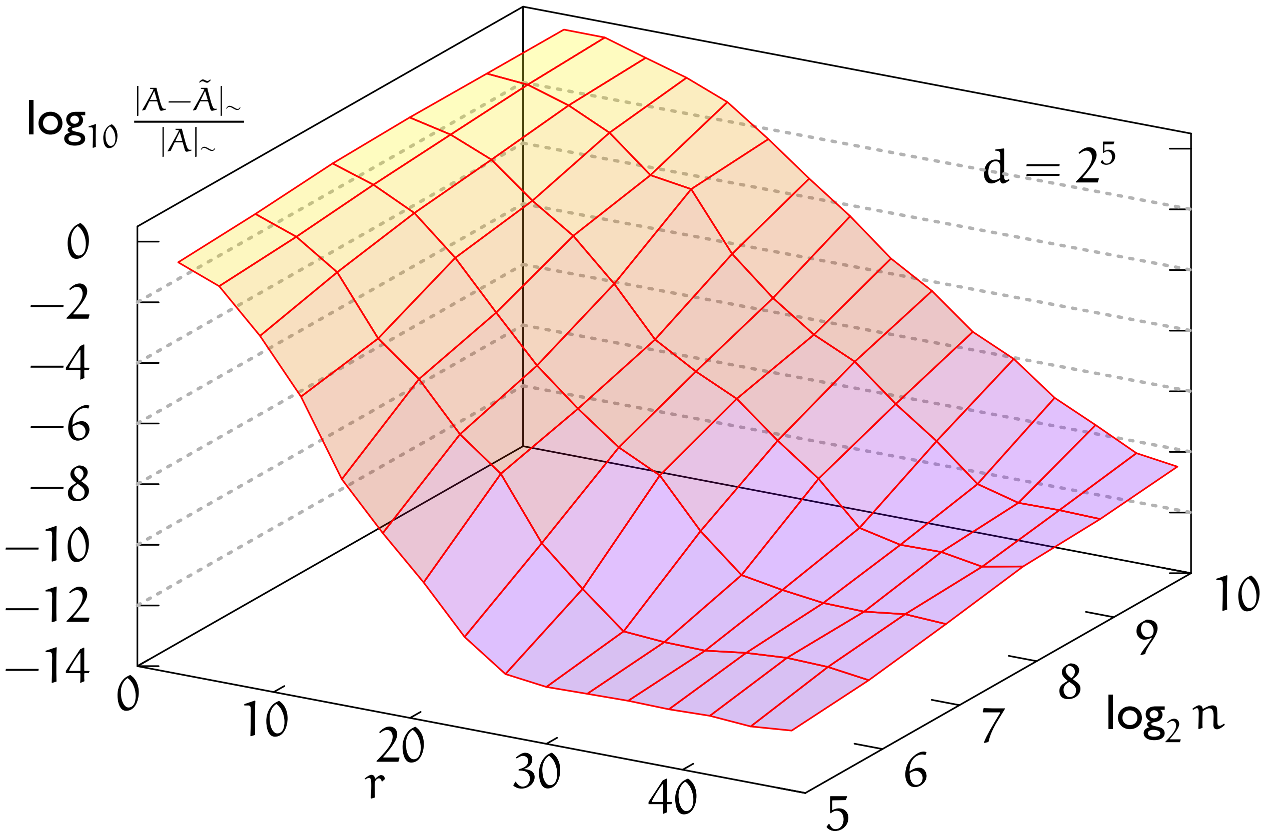
Runtime wrt rank for $A(i_1, \ldots, i_d) = 1/\sqrt{i_1^2 + \cdots + i_d^2}$



Scales quadratically

Accuracy of interpolation

$$A(i_1, \ldots, i_d) = (i_1^2 + \cdots + i_d^2)^{-\frac{1}{2}}, \qquad i_k = 1, \ldots, n, \quad k = 1, \ldots, d.$$



Reconstruct $n^d = (2^{10})^{32} = 2^{320} \approx 10^{100} =$ googol entries
from $dnr^2 = 32 \cdot 2^{10} \cdot 50^2 \approx 8 \cdot 10^7$ samples $= 1.2$ GB of data

Outline

- Introduction
- Cross interpolation theory
- Cross interpolation practice
  - — Cross interpolation algorithms for tensors
  - ⇒ High–dimensional integration
  - — Stochastic problems
  - — Parallel algorithm

Generalised Gaussian integral

$$\int_{\mathbb{R}^d} e^{-\frac{1}{2}x^T A x} dx = \frac{(2\pi)^{d/2}}{\sqrt{\det(A)}}$$

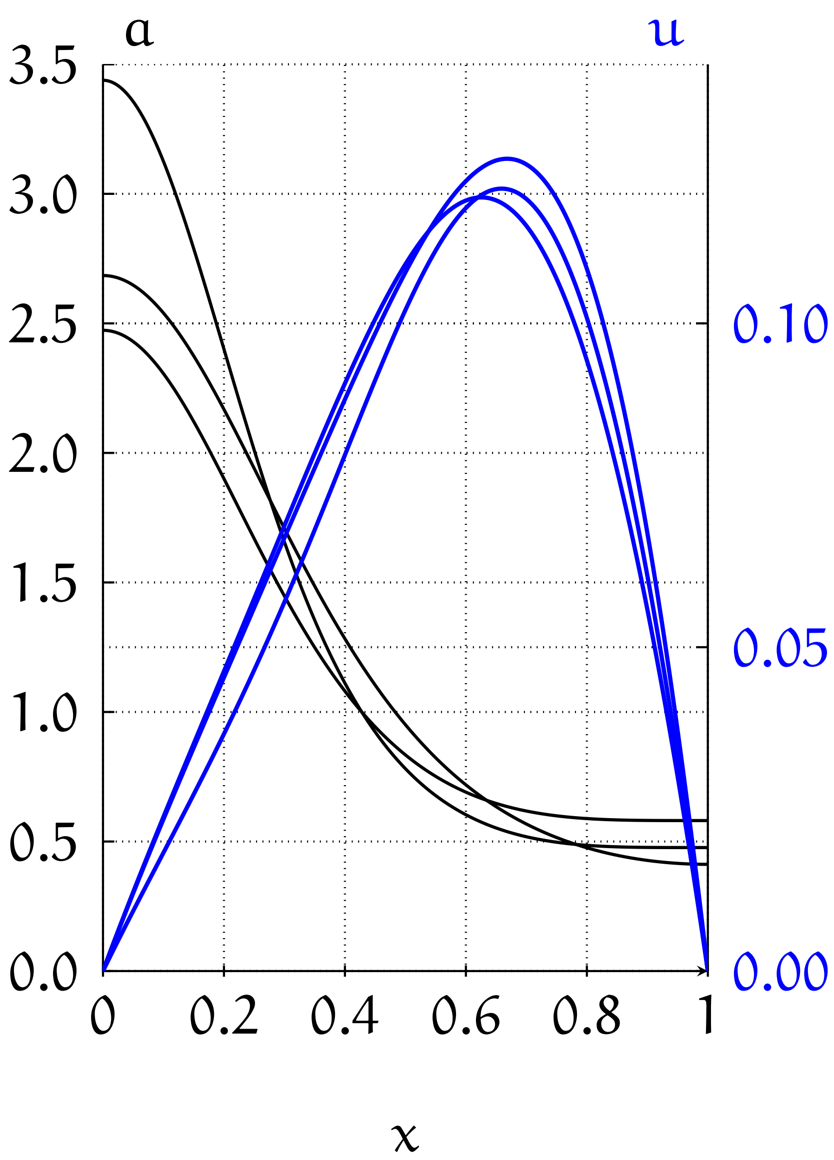
- $A = \left[e^{-|i-j|}\right]_{i,j=1}^{d}, \; d = 100, \; n = 50$ nodes in each mode
- $n^d = 50^{100}$ quadrature nodes overall — **curse of dimensionality**

$\log_{10} \frac{|\mathcal{J}-\tilde{\mathcal{J}}|}{\mathcal{J}}$
−1
−2
−3
−4
−5
−6
−7
−8
−9
−10
−11
−12
5
6
7
$\log_{10} N_{\text{eval}}$

Outline

- Introduction
- Cross interpolation theory
- Cross interpolation practice
  - — Cross interpolation algorithms for tensors
  - ⇒ Stochastic problems
  - — Parallel algorithm

Stochastic PDE



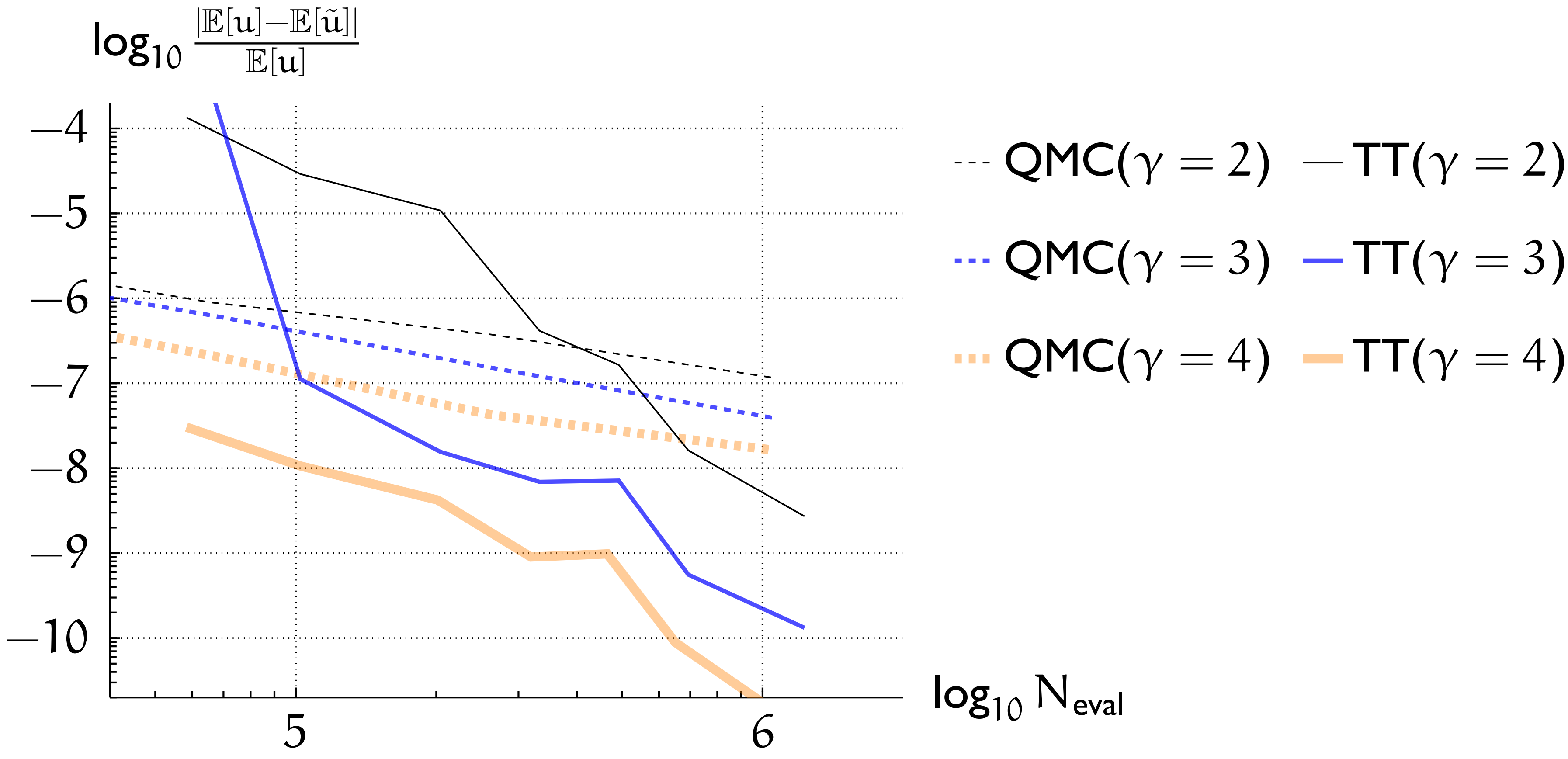
Karhunen-Loéve expansion

$$-\nabla_x(a(x,\xi)\nabla_x u(x,\xi)) = 1 \quad x \in [0,1]$$
$$u(x,\xi) = 0 \quad x \in \{0,1\}$$

$$a(x,\xi) = \exp\left(\sum_{k=1}^{d} \frac{1}{k^\gamma} \cos(k\pi x)\xi_k\right)$$

- $d \sim 10$ parameters $\xi_k \in [-1,1]$
- $n = 15$ LG quadrature points per ξ_k
- $n^d = 15^{10}$ PDEs to solve
- **curse of dimensionality**
- apply tensor formats

Stochastic PDE

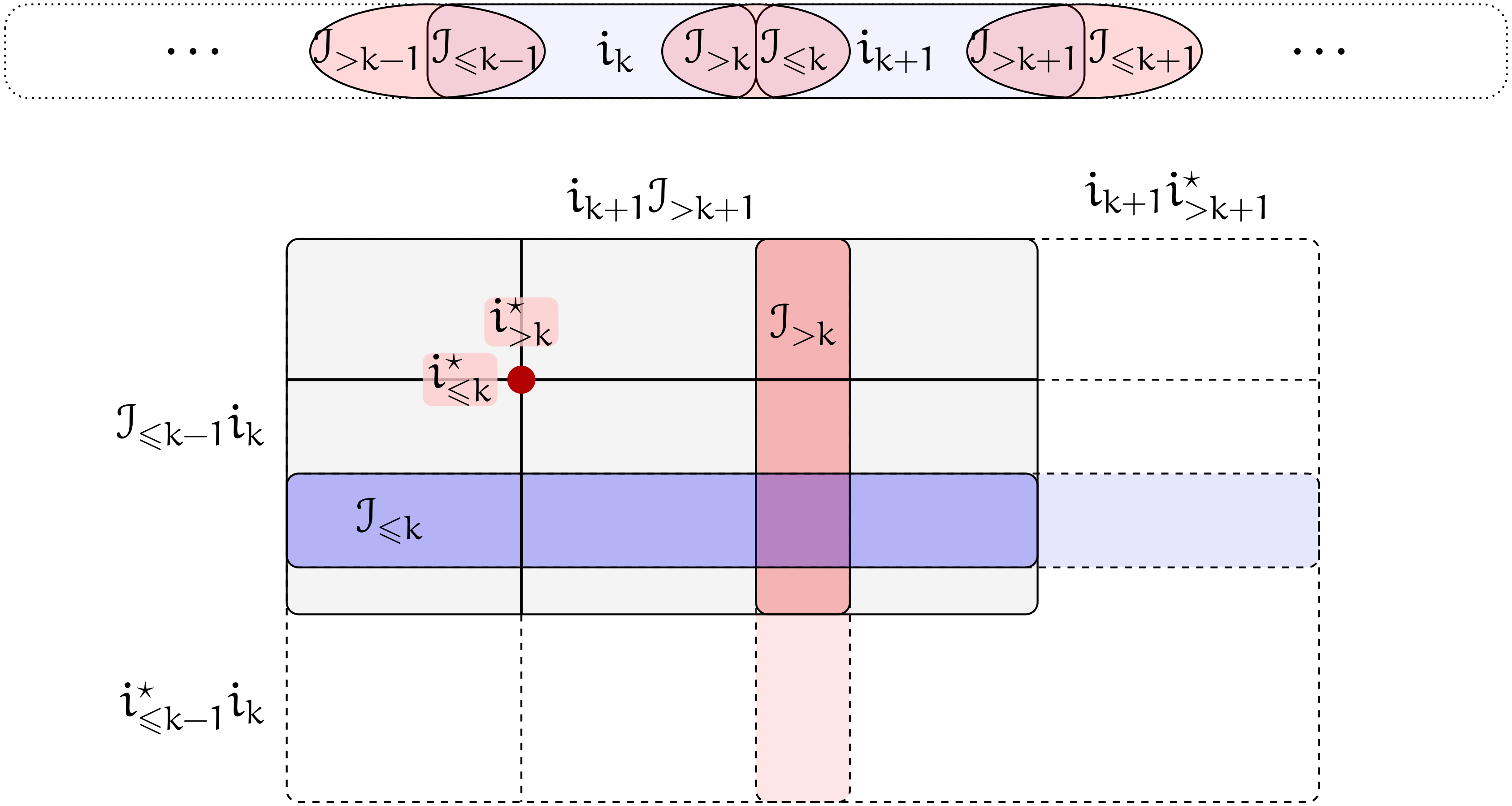


- TT-cross interpolation converges faster than Quasi Monte Carlo
- TT-cross interpolation likes smooth coefficients
- Quasi Monte Carlo still works better when low accuracy is required

Outline

- Introduction
- Cross interpolation theory
- Cross interpolation practice
  - — Cross interpolation algorithms for tensors
  - — Stochastic problems
  - ⇒ Parallel algorithm

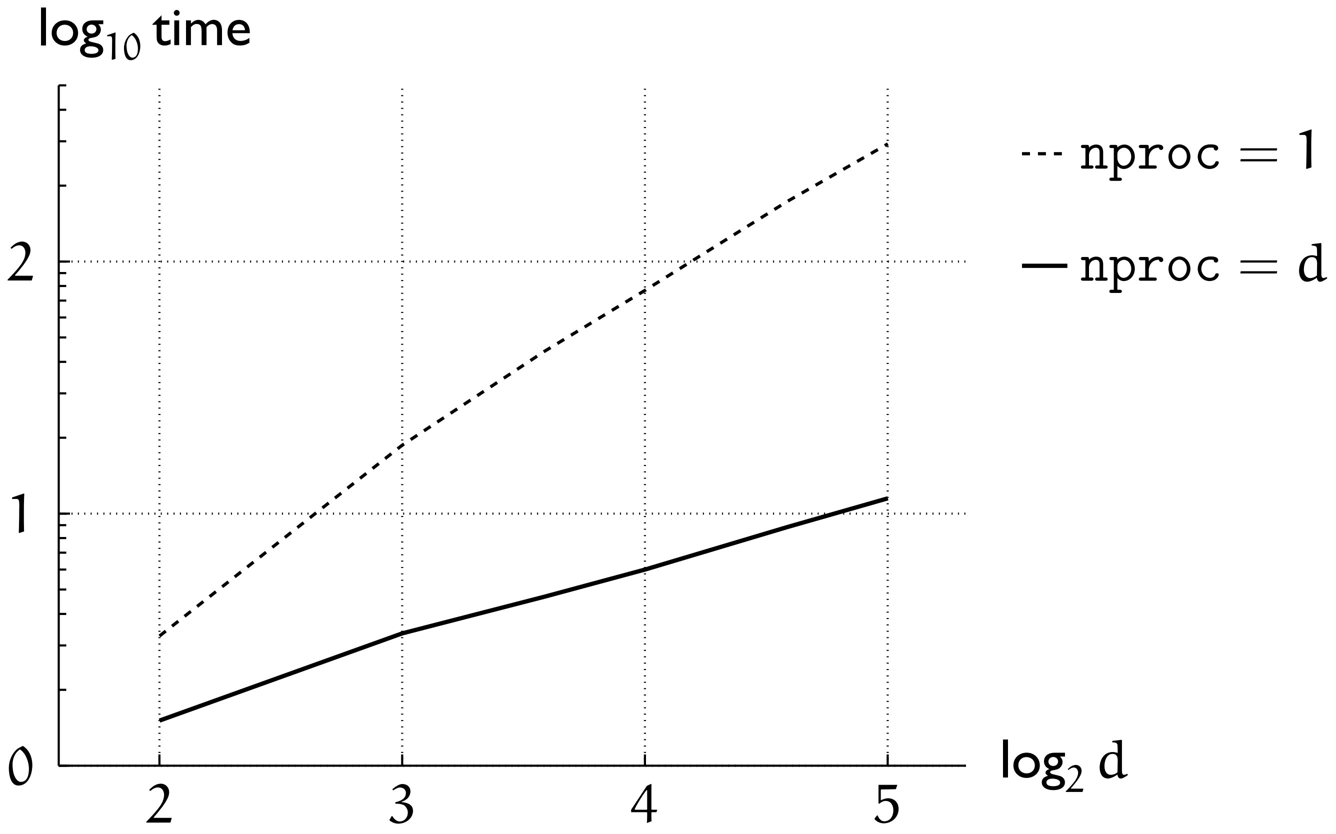
Parallel implementation



Related work

- [Grasedyck et al, Comput. Visual. Sci. 2015] Parallel random sampling in HT
- [Etter, SISC 2016] Parallel ALS for solving linear systems

Scalability



- implemented in Fortran+MPI
- sequential code scales quadratically with d
- parallel code scales linearly with d (each element costs $\sim d$)

Conclusion

- tensor interpolation is accurate (for maxvol sets)
- tensor interpolation formula interpolates (for nested sets)
- tensor interpolation algorithm is fast, scalable and reliable

References

- Quasioptimality of maximum-volume cross interpolation of tensors

  Linear Algebra and Applications 458:217–244, **2014**