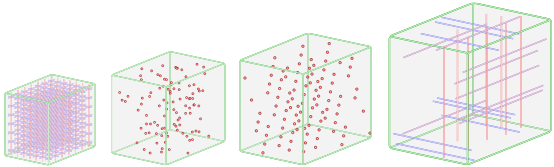
Low–rank interpolation of tensors for high–precision calculation of high–dimensional integrals

Dmitry Savostyanov

University of Brighton



Structured Matrix Days
Limoges, 24th May 2019

Why do we need high–dimensional integrals?

Integration in higher dimensions

Why do we need high–dimensional integrals?

- Multivariate distributions: $\int_{\mathbb{R}^d} f(x) \left(\sum_k b_k e^{-\frac{1}{2}(x-c_k)^T A_k (x-c_k)} \right) dx$
- Stochastic and parametric PDEs:

$$\mathcal{D}_x(\xi) u(x,\xi) = f(x), \qquad \text{find} \quad \bar{u}(x) = \mathsf{E}[u(\xi,x)]$$

Integration in higher dimensions

Why do we need high–dimensional integrals?

- Multivariate distributions: $\int_{\mathbb{R}^d} f(x) \left(\sum_k b_k e^{-\frac{1}{2}(x-c_k)^T A_k (x-c_k)} \right) dx$
- Stochastic and parametric PDEs:

$$\mathcal{D}_x(\xi) u(x,\xi) = f(x), \qquad \text{find} \quad \bar{u}(x) = \mathsf{E}[u(\xi,x)]$$

- Ising susceptibility integrals in mathematical physics
- Weyl's integrals in representation theory / random matrix theory

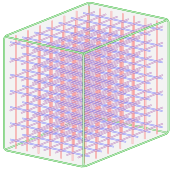
$$\int_{\mathbb{U}_d} f(U) dU = \frac{1}{(2\pi)^d d!} \int_{[0,2\pi]^d} f(\mathrm{diag}(e^{i\theta_1}, \ldots, e^{i\theta_d})) \prod_{j \leqslant k} \left| e^{i\theta_j} - e^{i\theta_k} \right|^2 d\theta$$

- ... and many more.

Integration in higher dimensions

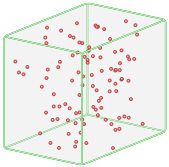
How can we compute high–dimensional integrals?

- Tensor product grid



bad

- Monte Carlo

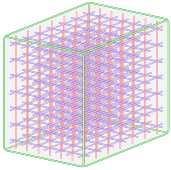


ok but slow

Integration in higher dimensions

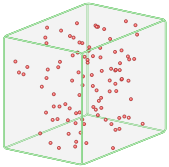
How can we compute high–dimensional integrals?

- Tensor product grid



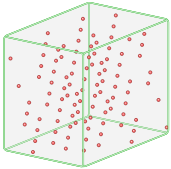
bad

- Monte Carlo



ok but slow

- quasi Monte Carlo

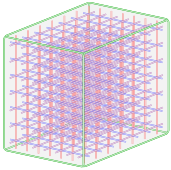


better

Integration in higher dimensions

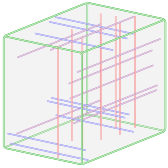
How can we compute high–dimensional integrals?

- Tensor product grid



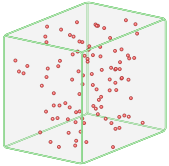
bad

- Cross interpolation



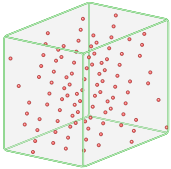
even better

- Monte Carlo



ok but slow

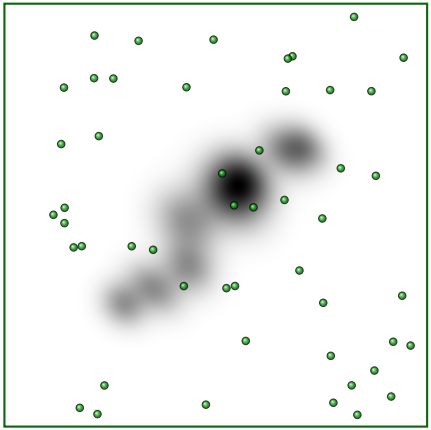
- quasi Monte Carlo



better

Example 1: Generalised Gaussians

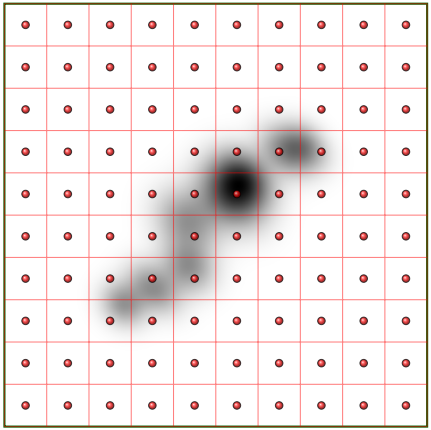
Example 1: Generalised Gaussians



- Monte Carlo

$$\varepsilon \sim N_{\text{eval}}^{-1/2}$$

Example 1: Generalised Gaussians



- Monte Carlo

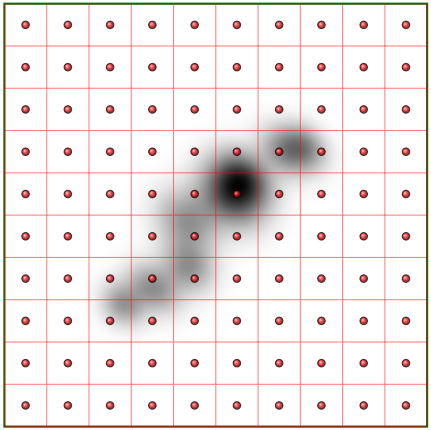
$$\varepsilon \sim N_{\text{eval}}^{-1/2}$$

- Tensor product grid

$$\varepsilon \sim n^{-s} \quad \text{but} \quad N_{\text{eval}} = n^d$$

Matrices: low–rank structure

Example 1: Generalised Gaussians



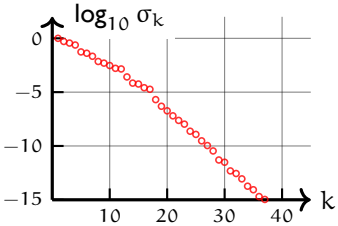
- Monte Carlo

$$\varepsilon \sim N_{\text{eval}}^{-1/2}$$

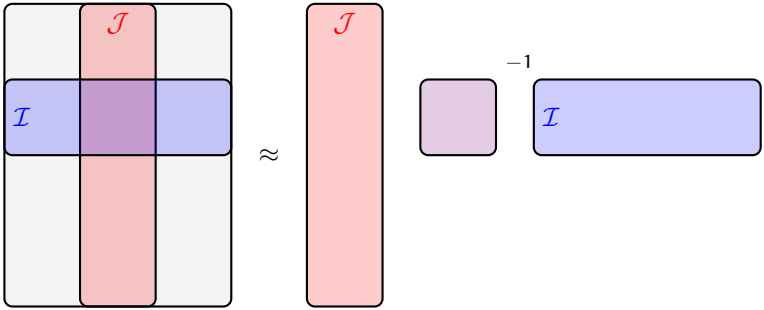
- Tensor product grid

$$\varepsilon \sim n^{-s} \quad \text{but} \quad N_{\text{eval}} = n^d$$

Low-rank structure of p.d.f.:



Matrices: cross interpolation



A rank–r approximation can be recovered from $\mathcal{O}(nr)$ elements:

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

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

[Goreinov, Tyrtyshnikov, Zamarashkin, 1995], [Tyrtyshnikov, 1996], [Tyrtyshnikov, 2000]

Matrices: Cross interpolation (good)

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

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

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

Matrices: Cross interpolation (not so good)

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

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

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

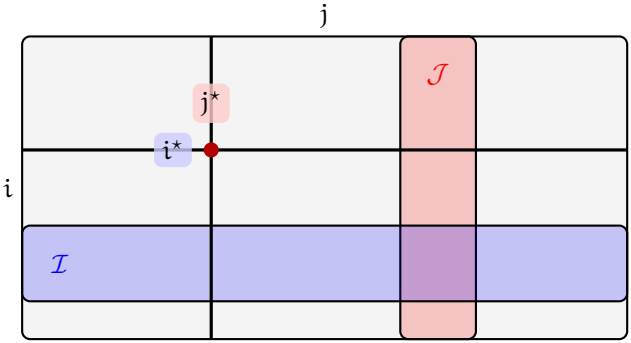
Matrices: Cross interpolation (maximum volume)

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

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

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

Matrices: Cross interpolation algorithm



Algorithm (Gaussian elimination with partial pivoting)

- Find $(i^\star, j^\star)$ s.t. $\left|A(i^\star, j^\star) - A(i^\star, \mathcal{J})[A(\mathcal{I}, \mathcal{J})]^{-1} A(\mathcal{I}, j^\star)\right|$ is large
- Add $i^\star$ to $\mathcal{I}$ and $j^\star$ to $\mathcal{J}$
- Update columns $[A(i, \mathcal{J})]$, submatrix $[A(\mathcal{I}, \mathcal{J})]^{-1}$, and rows $[A(\mathcal{I}, j)]$

Matrices: Cross interpolation algorithm (step 1)

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

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

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

Matrices: Cross interpolation algorithm (step 1)

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

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

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

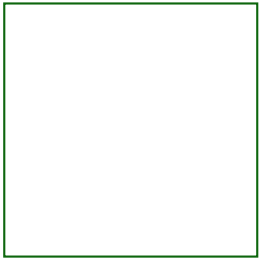
Matrices: examples

Example 1: Generalised Gaussians

matrix

residual, rank 0

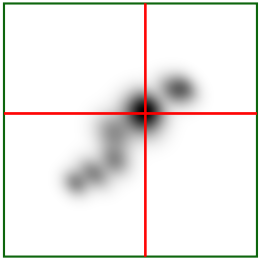
interpolation, rank 0



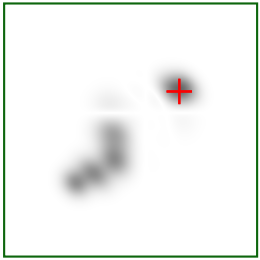
Matrices: examples

Example 1: Generalised Gaussians

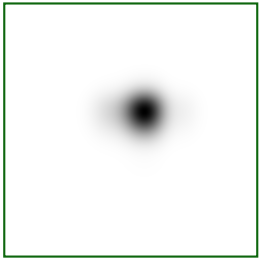
matrix



residual, rank 1



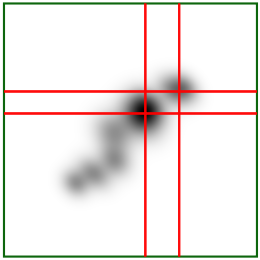
interpolation, rank 1



Matrices: examples

Example 1: Generalised Gaussians

matrix



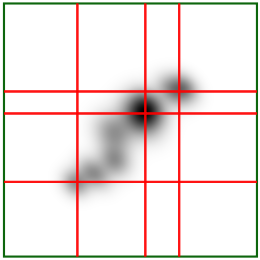
residual, rank 2

interpolation, rank 2

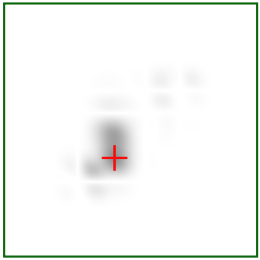
Matrices: examples

Example 1: Generalised Gaussians

matrix



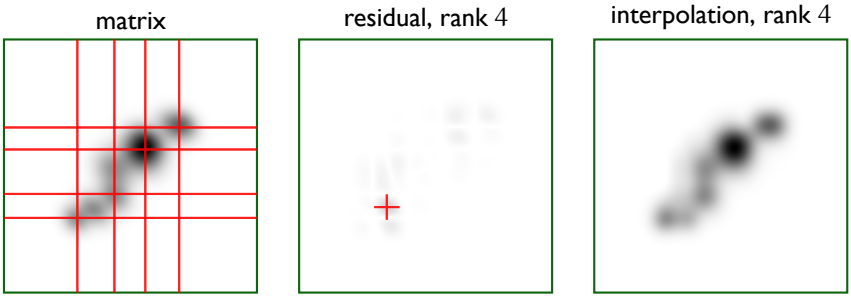
residual, rank 3



interpolation, rank 3

Matrices: examples

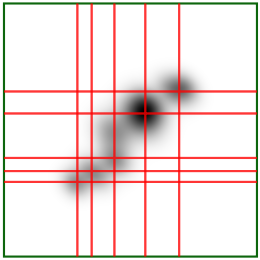
Example 1: Generalised Gaussians



Matrices: examples

Example 1: Generalised Gaussians

matrix



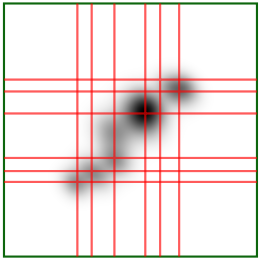
residual, rank 5

interpolation, rank 5

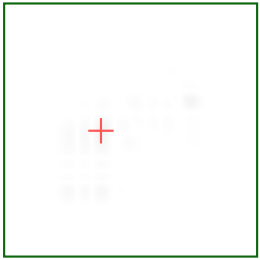
Matrices: examples

Example 1: Generalised Gaussians

matrix



residual, rank 6



interpolation, rank 6

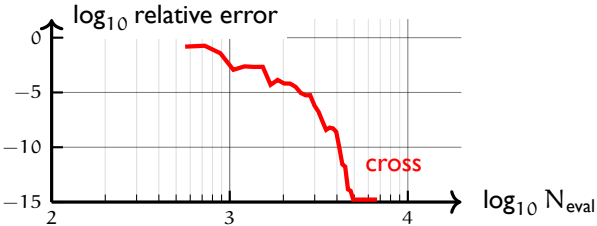
Matrices: examples

Example 1: Generalised Gaussians

matrix

residual, rank 7

interpolation, rank 7



Tensors: formats

Tensor product formats for $f(x_1, x_2, \cdots, x_d)$

canonical polyadic (CP)

$$\sum_k u_k(x_1) v_k(x_2) \cdots w_k(x_d)$$

Applications to Independent component analysis (ICA) for clustering, blind source separation (BSS) for DOA, EEG, chemometrics, ...

Tensor train (TT)

$$\sum_{\mathbf{k}} u_{k_1}(x_1) v_{k_1,k_2}(x_2) \cdots w_{k_{d-1}}(x_d)$$

A general tool for compression of high–dimensional data. Applications to solution of high–dimensional linear problems, sampling from high–dimensional probability distributions, control of high–dimensional dynamic systems,...

Hierarchical Tucker (HT)

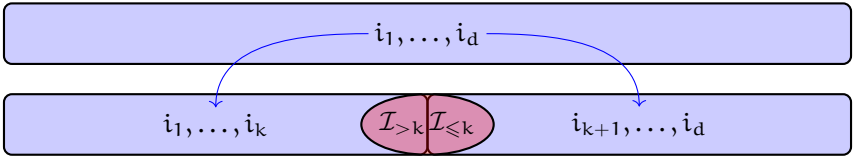
More general than TT

Harder to develop

Tensors: cross interpolation

Matrix interpolation for tensors

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

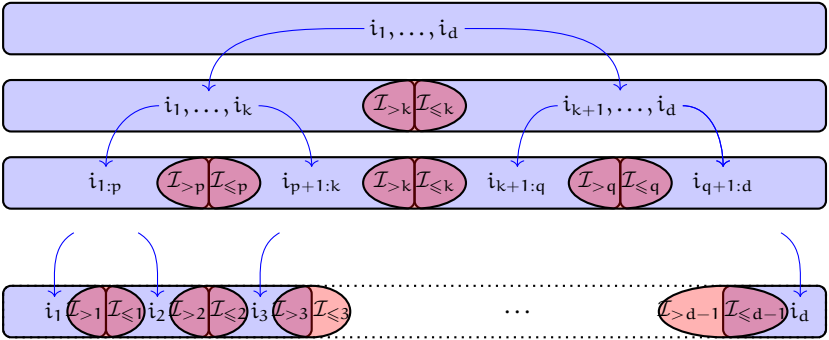


If $[\mathcal{I}_{\leqslant k}, \mathcal{I}_{>k}] = \mathrm{maxvol}[A(i_{\leqslant k}, i_{>k})]$, then

$$|A - \tilde{A}| \leqslant (r_k + 1)^2 \min_{\operatorname{rank} X = r_k} |A - X|$$

[Goreinov, Tyrtyshnikov, 2011]

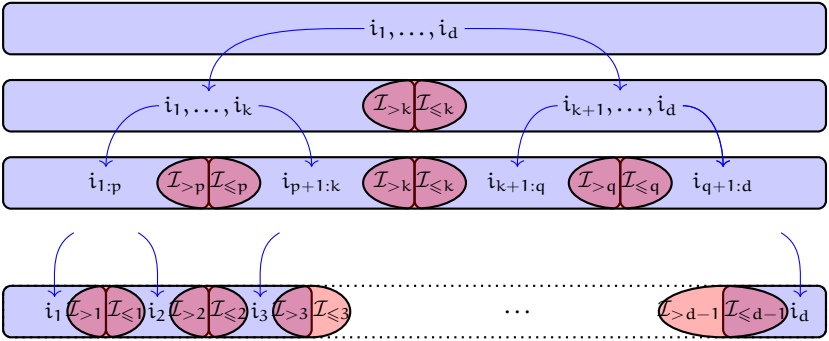
Tensors: cross interpolation



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

[Oseledets, Tyrtyshnikov, 2010] [Savostyanov, Oseledets, 2011]

Tensors: cross interpolation



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

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

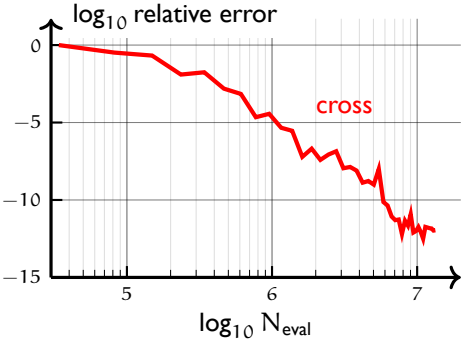
where $r = \max r_k, \quad \kappa = \max \kappa_k, \quad \kappa_k = r_k |A| |A_k^{-1}|$ [Savostyanov, 2014]

Tensors: examples

Example 1: Generalised Gaussian

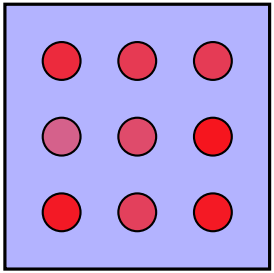
$$\int_{\mathbb{R}^d} e^{-\frac{1}{2}x^T A x}\, dx = \frac{(2\pi)^{d/2}}{\sqrt{\det(A)}}$$

- $A = \left[e^{-|i-j|}\right]_{i,j=1}^{n}$
- $d = 100$, $n = 50$



Tensors: examples

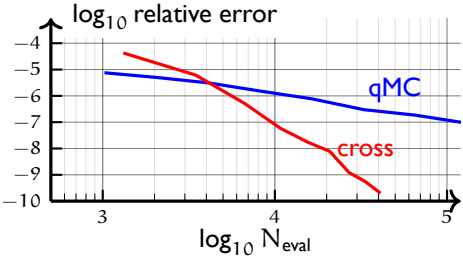
Example 2: Parametric PDE



$$-\nabla_x(a(x,p)\nabla_x u(x,p)) = 1 \quad x \in \Omega$$
$$u(x,p) = 0 \quad x \in \partial\Omega$$

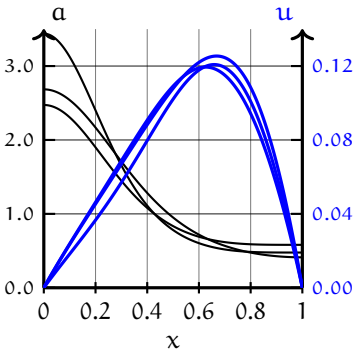
$$a(x,p) = \begin{cases} p_{s,t} & x \in \text{cookie}_{s,t} \\ 1 & \text{otherwise} \end{cases}$$

- $p_{s,t} \in [\frac{1}{2}, 2]$
- find $E[\int_\Omega u\, dx]$



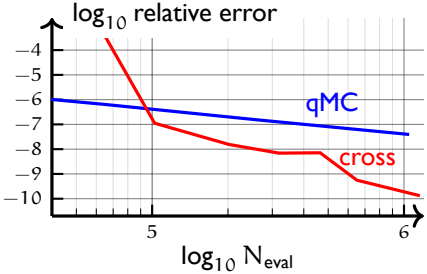
Tensors: examples

Example 3: Stochastic ODE



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

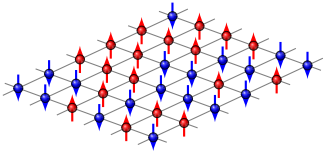
$$a(x,\xi) = \exp\left(\sum_{k=1}^{d} k^{-\gamma}\cos(k\pi x)\xi_k\right)$$



- $\xi_k \in [-1,1]$, $d \sim 10$
- Find $E[\int_0^1 u(x,\xi)\,dx]$

Ising susceptibility integrals

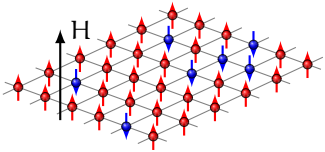
Example 4: Ising model



- electrons have spins: $|\uparrow\rangle$ or $|\downarrow\rangle$

Ising susceptibility integrals

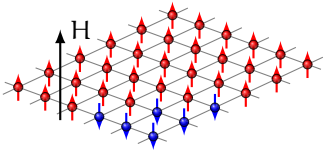
Example 4: Ising model



- electrons have spins: $|\uparrow\rangle$ or $|\downarrow\rangle$
- external magnetic field aligns the spins

Ising susceptibility integrals

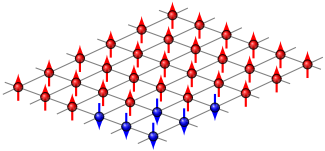
Example 4: Ising model



- electrons have spins: $|\uparrow\rangle$ or $|\downarrow\rangle$
- external magnetic field aligns the spins
- in ferromagnets, spins form domains

Ising susceptibility integrals

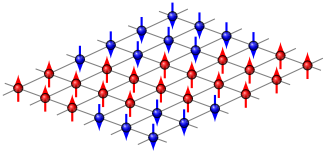
Example 4: Ising model



- electrons have spins: $|\uparrow\rangle$ or $|\downarrow\rangle$
- external magnetic field aligns the spins
- in ferromagnets, spins form domains
- domains persist even when the external field is zero

Ising susceptibility integrals

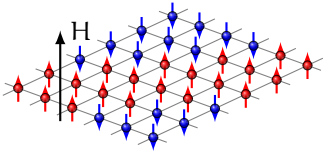
Example 4: Ising model



- electrons have spins: $|\uparrow\rangle$ or $|\downarrow\rangle$
- external magnetic field aligns the spins
- in ferromagnets, spins form domains
- domains persist even when the external field is zero
- heating and/or beating removes the total magnetisation, but domains still form *spontaneously*

Ising susceptibility integrals

Example 4: Ising model



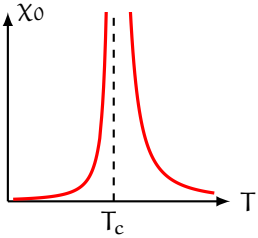
- electrons have spins: $|\uparrow\rangle$ or $|\downarrow\rangle$
- external magnetic field aligns the spins
- in ferromagnets, spins form domains
- domains persist even when the external field is zero
- heating and/or beating removes the total magnetisation, but domains still form *spontaneously*

- Systems with next–neighbour interaction exhibit co-operative behavior (similar to gas–liquid transition, binary alloys, biology, genetics, economics, etc)
- Phase transition effect: $\begin{cases} \text{spontaneous magnetisation,} & \text{when } T < T_c \\ \text{demagnetisation,} & \text{when } T > T_c \end{cases}$
- Susceptibility $\chi_0(T) = -\left.\dfrac{\partial^2 f}{\partial H^2}\right|_{H=0}$ is closely related to the long–range correlation $\langle \hat{\sigma}_{0,0}\, \hat{\sigma}_{m,n} \rangle$

Ising susceptibility integrals

- Phase transition at Curie temperature:

$$\chi_0^{\pm}(T) \sim C_0^{\pm}\,|1 - T/T_c|^{-7/4}$$



- Susceptibility amplitudes

$$C_0^{+} \sim \sum_{d \text{ odd}} \frac{\pi D_d}{(2\pi)^d}, \quad C_0^{-} \sim \sum_{d \text{ even}} \frac{\pi D_d}{(2\pi)^d},$$

$$D_d = \int_{\square} \frac{\prod_{1 \leqslant i < j \leqslant d} \left(\frac{1 - x_{i+1} \cdots x_j}{1 + x_{i+1} \cdots x_j}\right)^2 dx_2 \cdots dx_d}{\left(1 + \sum_{k=2}^{d} x_2 \cdots x_k\right)\left(1 + \sum_{k=2}^{d} x_k \cdots x_d\right)}$$

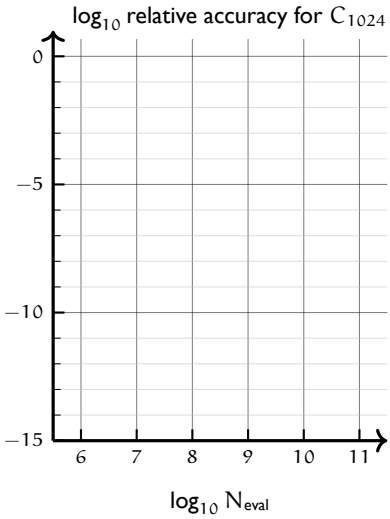
[Wu, McCoy, Tracy, Barouch, 1976]

$$C_d = \int_{\square} \frac{1}{\left(1 + \sum_{k=2}^{d} x_2 \cdots x_k\right)\left(1 + \sum_{k=2}^{d} x_k \cdots x_d\right)}$$

[Bailey, Borwein, Crandall, 2006]

Ising susceptibility integrals

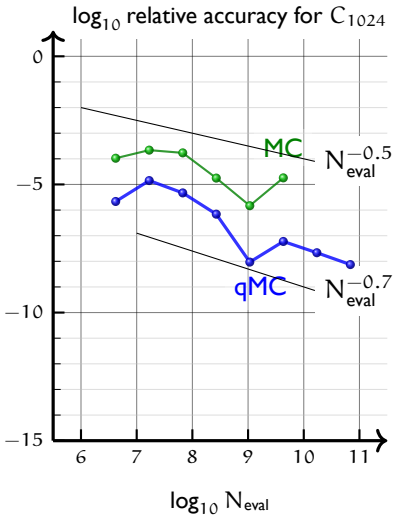
Verification



- C_{1024} is reduced to two–dimensional integral and computed to 500 digits [Bailey, Borwein, Crandall, 2006]
- We compute C_{1024} as 1023–dimensional integral to verify cross interpolation

Ising susceptibility integrals

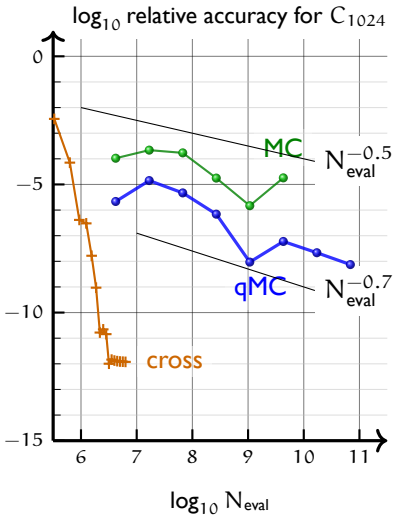
Verification



- C_{1024} is reduced to two–dimensional integral and computed to 500 digits [Bailey, Borwein, Crandall, 2006]
- We compute C_{1024} as 1023–dimensional integral to verify cross interpolation
- MC and qMC converge slowly

Ising susceptibility integrals

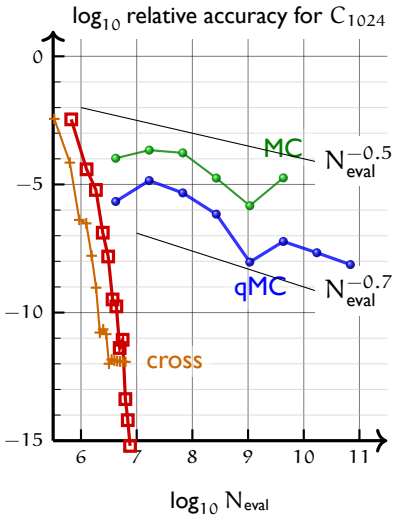
Verification



- C_{1024} is reduced to two–dimensional integral and computed to 500 digits [Bailey, Borwein, Crandall, 2006]
- We compute C_{1024} as 1023–dimensional integral to verify cross interpolation
- MC and qMC converge slowly
- TT cross interpolation
  - + — double precision

Ising susceptibility integrals

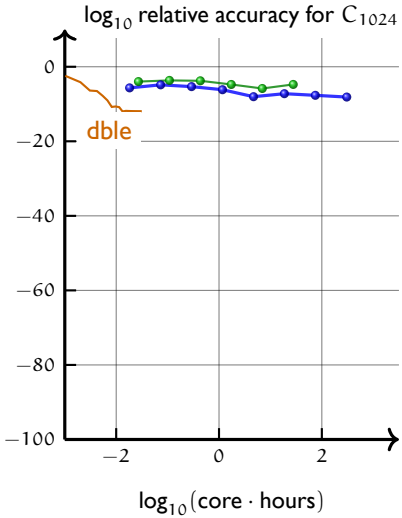
Verification



- C_{1024} is reduced to two–dimensional integral and computed to 500 digits
  [Bailey, Borwein, Crandall, 2006]
- We compute C_{1024} as 1023–dimensional integral to verify cross interpolation
- MC and qMC converge slowly
- TT cross interpolation
  - + — double precision
  - □ — quadruple precision

Ising susceptibility integrals

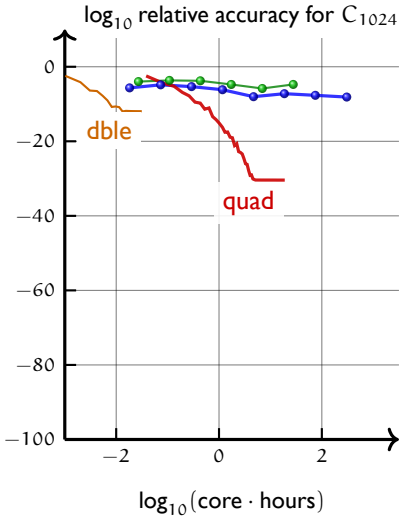
Benchmarking



- High–precision numerics is more expensive — it's fair to compare CPU time.

Ising susceptibility integrals

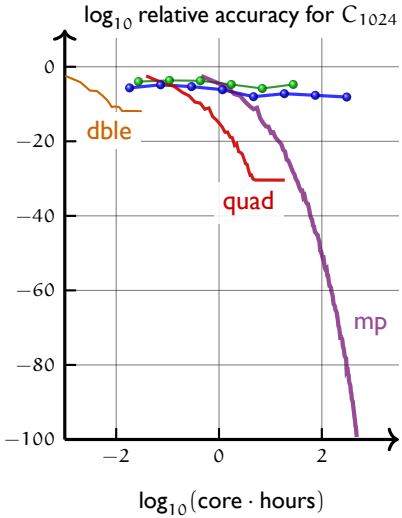
Benchmarking



- High–precision numerics is more expensive — it's fair to compare CPU time.
- Quadruple precision via `gfortran`'s option `default-real-8`

Ising susceptibility integrals

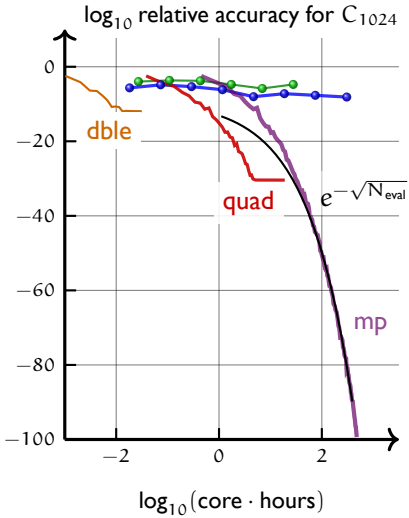
Benchmarking



- High–precision numerics is more expensive — it's fair to compare CPU time.
- Quadruple precision via `gfortran`'s option `default-real-8`
- Multiple precision via `MPFUN2015` package [D. H. Bailey]

Ising susceptibility integrals

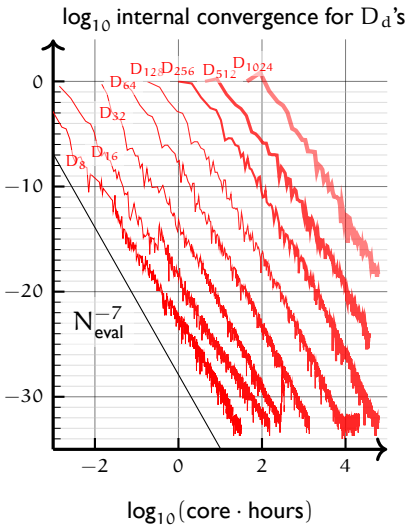
Benchmarking



- High–precision numerics is more expensive — it's fair to compare CPU time.
- Quadruple precision via `gfortran`'s option `default-real-8`
- Multiple precision via `MPFUN2015` package [D. H. Bailey]
- Is it exponential convergence we see?

Ising susceptibility integrals

Exploration



- Each element of D_n cost $\mathcal{O}(n^2)$ which is n times more expensive than C_n
- For D_{1024} can get 18 digits using 4 days on 512 CPU nodes $\sim 10^5$ corehours, $\sim 10^3$ kWh, $\sim$ 400£
- Good news: the observed convergence of TT cross interpolation is $\mathcal{O}(N^{-7})$
- Other news: we are still far from the 100 digit target that enables us to use inverse symbolic calculators

Conclusions

- tensor cross interpolation is fast, scalable and reliable
- tensor cross interpolation allows high–precision integration
- tensor cross interpolation can replace Monte Carlo methods (we hope)

References

- Quasioptimality of maximum-volume cross interpolation of tensors. Linear Algebra and Applications 458:217–244, 2014.
- Parallel cross interpolation for high–precision calculation of high–dimensional integrals. ArXiv:1903.11554