MAX PLANCK GESELLSCHAFT

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

A hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling

Bonan Sun (Max Planck Institute Magdeburg)

GAMM Annual Meeting, April 8, 2024

Based on a joint work with
Peter Benner (MPI Magdeburg)
Venera Khoromskaia (MPI MiS)
Boris Khoromskij (MPI MiS)

Partners:

MAX PLANCK INSTITUTE FOR MATHEMATICS IN THE SCIENCES

Organization

1. ChebTuck format introduction
2. Numerical schemes for ChebTuck approximation
3. Applications to multi-particle modelling

Algebraic low-rank tensor approximations

- Consider $f : [-1,1]^d \to \mathbb{R}$, $d = 3$.
- Goal: approximate f with a *small number of parameters* $\rightsquigarrow$ cheap computations with f.

Algebraic low-rank tensor approximations

- Consider $f : [-1, 1]^d \to \mathbb{R}$, $d = 3$.
- Goal: approximate f with a *small number of parameters* $\rightsquigarrow$ cheap computations with f.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, e.g., Tucker format:

$$\mathbf{F} \approx \sum_{i_1, i_2, i_3 = 1}^{r_1, r_2, r_3} \beta_{i_1, i_2, i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

Algebraic low-rank tensor approximations

- Consider $f : [-1,1]^d \to \mathbb{R}$, $d = 3$.
- Goal: approximate f with a *small number of parameters* $\rightsquigarrow$ cheap computations with f.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, e.g., Tucker format:

$$\mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

- e.g., $\mathbf{F}$ contains the function values:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}),\ \ t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h_\ell,\ \ i_\ell = 1, \cdots, n_\ell,$$

or $\mathbf{F}$ contains projection of f on some basis functions

$$\mathbf{F}_{i_1,i_2,i_3} = \int_{[-1,1]^3} f(x_1,x_2,x_3)\phi_{i_1}^{(1)}(x_1)\phi_{i_2}^{(2)}(x_2)\phi_{i_3}^{(3)}(x_3)\,\mathrm{d}\,x_1\,\mathrm{d}\,x_2\,\mathrm{d}\,x_3.$$

Algebraic low-rank tensor approximations

- Consider $f : [-1, 1]^d \to \mathbb{R}$, $d = 3$.
- Goal: approximate f with a *small number of parameters* $\rightsquigarrow$ cheap computations with f.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, e.g., Tucker format:

$$\mathbf{F} \approx \sum_{i_1, i_2, i_3 = 1}^{r_1, r_2, r_3} \beta_{i_1, i_2, i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

- Storage: $\mathcal{O}(dnr + r^d)$, $n = \max\{n_1, n_2, n_3\}$, $r = \max\{r_1, r_2, r_3\}$.

Algebraic low-rank tensor approximations

- Consider $f : [-1,1]^d \to \mathbb{R}$, $d = 3$.
- Goal: approximate f with a *small number of parameters* $\rightsquigarrow$ cheap computations with f.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, e.g., Tucker format:

$$\mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

- Storage: $\mathcal{O}(dnr + r^d)$, $n = \max\{n_1, n_2, n_3\}$, $r = \max\{r_1, r_2, r_3\}$.
- Disadvantage: *large* n required to achieve high accuracy.

Algebraic low-rank tensor approximations

- Consider $f : [-1,1]^d \to \mathbb{R}$, $d = 3$.
- Goal: approximate f with a *small number of parameters* $\rightsquigarrow$ cheap computations with f.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, e.g., Tucker format:

$$\mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

- Storage: $\mathcal{O}(dnr + r^d)$, $n = \max\{n_1, n_2, n_3\}$, $r = \max\{r_1, r_2, r_3\}$.
- Disadvantage: *large* n required to achieve high accuracy.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format.

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

Storage: $\mathcal{O}(dnr + r^d)$.

Functional low-rank tensor approximations

- Consider $f : [-1, 1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1, i_2, i_3 = 1}^{r_1, r_2, r_3} \beta_{i_1, i_2, i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1,x_2,x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \; \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

- Various operations can be performed efficiently, e.g., integration:

$$\int_{[-1,1]^3} f(x_1,x_2,x_3)\,\mathrm{d}\,\mathbf{x} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} \int_{-1}^{1} v_{i_1}^{(1)}(x_1)\,\mathrm{d}\,x_1 \int_{-1}^{1} v_{i_2}^{(2)}(x_2)\,\mathrm{d}\,x_2 \int_{-1}^{1} v_{i_3}^{(3)}(x_3)\,\mathrm{d}\,x_3$$

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

- Proper parameterization of $v_{i_\ell}^{(\ell)}(x_\ell)$

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

- Proper parameterization of $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell), \ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$.

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \; \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

- Proper parameterization of $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell), \; T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$.
- Storage: $\mathcal{O}(dmr + r^d)$. $m \ll n$ for the same accuracy as grid-based methods.

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1,x_2,x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

- Proper parameterization of $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell), \ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$.
- Storage: $\mathcal{O}(dmr + r^d)$. $m \ll n$ for the same accuracy as grid-based methods.
- We call it **ChebTuck format**. It is also the format Chebfun3 [Hashemi/Trefethen'17] assumes.

Functional low-rank tensor approximations

- Consider $f : [-1,1]^3 \to \mathbb{R}$. Goal: approximate f with a *small number of parameters*.
- **Grid-based methods**: (algebraic) low-rank tensor approximation of function related tensor

$$\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \ \mathbf{F} \approx \sum_{i_1, i_2, i_3 = 1}^{r_1, r_2, r_3} \beta_{i_1, i_2, i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)}.$$

  Storage: $\mathcal{O}(dnr + r^d)$.
- **Mesh-free methods**: functional low-rank tensor approx. $\rightsquigarrow$ focus on Tucker format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1, i_2, i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

- Proper parameterization of $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell, i_\ell}^{(\ell)} T_{j_\ell - 1}(x_\ell), \ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$.
- Storage: $\mathcal{O}(dmr + r^d)$. $m \ll n$ for the same accuracy as grid-based methods.
- We call it **ChebTuck format**. It is also the format Chebfun3 [Hashemi/Trefethen'17] assumes.
- Goal: given f, compute its ChebTuck approximation.

Organization

ChebTuck approximation: functional case

- Goal: approximate f in ChebTuck format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3), \;\; v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell).$$

ChebTuck approximation: functional case

- Goal: approximate f in ChebTuck format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3), \;\; v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell).$$

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1, x_2, x_3) \approx \tilde{f}_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

ChebTuck approximation: functional case

- Goal: approximate f in ChebTuck format:

$$f(x_1, x_2, x_3) \approx \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3), \;\; v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell).$$

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1, x_2, x_3) \approx \tilde{f}_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ contains the Chebyshev coefficients

ChebTuck approximation: functional case

- Goal: approximate f in ChebTuck format:

$$f(x_1,x_2,x_3) \approx \sum_{i_1=1}^{r_1}\sum_{i_2=1}^{r_2}\sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3), \;\; v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell).$$

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1,x_2,x_3) \approx \tilde{f}_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{m_1}\sum_{i_2=1}^{m_2}\sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1\times m_2\times m_3}$ contains the Chebyshev coefficients computed by

$$\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} \text{ for some DCT matrices } W^{(\ell)}$$

$$\mathbf{T}_{i_1,i_2,i_3} = f(s_{i_1}^{(1)}, s_{i_2}^{(2)}, s_{i_3}^{(3)}),\; s_{i_\ell}^{(\ell)} = \cos((i_\ell-1)\pi/(m_\ell-1))$$

ChebTuck approximation: functional case

- Goal: approximate f in ChebTuck format:

$$f(x_1,x_2,x_3) \approx \sum_{i_1=1}^{r_1}\sum_{i_2=1}^{r_2}\sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3), \quad v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell).$$

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1,x_2,x_3) \approx \tilde{f}_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{m_1}\sum_{i_2=1}^{m_2}\sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1\times m_2\times m_3}$ contains the Chebyshev coefficients computed by

$$\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} \text{ for some DCT matrices } W^{(\ell)}$$

$$\mathbf{T}_{i_1,i_2,i_3} = f(s_{i_1}^{(1)}, s_{i_2}^{(2)}, s_{i_3}^{(3)}),\ s_{i_\ell}^{(\ell)} = \cos((i_\ell-1)\pi/(m_\ell-1))$$

- Compute the Tucker approximation of $\mathbf{C}$: $\mathbf{C} \approx \hat{\mathbf{C}} = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$

ChebTuck approximation: functional case

- Goal: approximate f in ChebTuck format:

$$f(x_1,x_2,x_3) \approx \sum_{i_1=1}^{r_1}\sum_{i_2=1}^{r_2}\sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3), \;\; v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell).$$

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1,x_2,x_3) \approx \tilde{f}_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{m_1}\sum_{i_2=1}^{m_2}\sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1\times m_2\times m_3}$ contains the Chebyshev coefficients computed by

$$\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} \text{ for some DCT matrices } W^{(\ell)}$$

$$\mathbf{T}_{i_1,i_2,i_3} = f(s_{i_1}^{(1)}, s_{i_2}^{(2)}, s_{i_3}^{(3)}),\; s_{i_\ell}^{(\ell)} = \cos((i_\ell-1)\pi/(m_\ell-1))$$

- Compute the Tucker approximation of $\mathbf{C}$: $\mathbf{C} \approx \hat{\mathbf{C}} = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$
- Plugging $\hat{\mathbf{C}}$ in $\tilde{f}_{\mathbf{m}}$ gives the ChebTuck approximation.

ChebTuck approximation: functional case

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1, x_2, x_3) \approx \tilde{f}_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ contains the Chebyshev coefficients computed by

$$\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} \text{ for some DCT matrices } W^{(\ell)}$$

$$\mathbf{T}_{i_1,i_2,i_3} = f(s_{i_1}^{(1)}, s_{i_2}^{(2)}, s_{i_3}^{(3)}),\ s_{i_\ell}^{(\ell)} = \cos((i_\ell - 1)\pi/(m_\ell - 1))$$

- Compute the Tucker approximation of $\mathbf{C}$: $\mathbf{C} \approx \hat{\mathbf{C}} = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$
- Plugging $\hat{\mathbf{C}}$ in $\tilde{f}_{\mathbf{m}}$ gives the ChebTuck approximation.

ChebTuck approximation: functional case

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1, x_2, x_3) \approx \tilde{f}_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ contains the Chebyshev coefficients computed by

$$\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} \text{ for some DCT matrices } W^{(\ell)}$$

$$\mathbf{T}_{i_1,i_2,i_3} = f(s_{i_1}^{(1)}, s_{i_2}^{(2)}, s_{i_3}^{(3)}),\ s_{i_\ell}^{(\ell)} = \cos((i_\ell - 1)\pi/(m_\ell - 1))$$

- Compute the Tucker approximation of $\mathbf{C}$: $\mathbf{C} \approx \hat{\mathbf{C}} = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$
- Plugging $\hat{\mathbf{C}}$ in $\tilde{f}_{\mathbf{m}}$ gives the ChebTuck approximation.
- ChebTuck approximation $\rightsquigarrow$ reduced to Tucker decomposition.

ChebTuck approximation: functional case

- Start from multivariate Chebyshev interpolant $\tilde{f}_{\mathbf{m}}$ of f:

$$f(x_1, x_2, x_3) \approx \tilde{f}_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} \mathbf{C}_{i_1,i_2,i_3} T_{i_1-1}(x_1) T_{i_2-1}(x_2) T_{i_3-1}(x_3),$$

- where $\mathbf{C} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ contains the Chebyshev coefficients computed by

$$\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} \text{ for some DCT matrices } W^{(\ell)}$$

$$\mathbf{T}_{i_1,i_2,i_3} = f(s^{(1)}_{i_1}, s^{(2)}_{i_2}, s^{(3)}_{i_3}),\ s^{(\ell)}_{i_\ell} = \cos((i_\ell - 1)\pi/(m_\ell - 1))$$

- Compute the Tucker approximation of $\mathbf{C}$: $\mathbf{C} \approx \hat{\mathbf{C}} = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$
- Plugging $\hat{\mathbf{C}}$ in $\tilde{f}_{\mathbf{m}}$ gives the ChebTuck approximation.
- ChebTuck approximation $\rightsquigarrow$ reduced to Tucker decomposition.
- Many algorithms available: [Hashemi/Trefethen'17] & [Dolgov/Kressner/Stroessner'21] applied vairants of Cross3D [Rakhuba/Oseledets'15] to $\mathbf{T}$.

ChebTuck approximation: CP tensor input

- In many applications, f not given explicitly in the full domain $[-1, 1]^3$

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

ChebTuck approximation: CP tensor input

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}),\ t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

ChebTuck approximation: CP tensor input

- In many applications, f not given explicitly in the full domain $[-1, 1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}),\ t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}),\ t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1,\cdots,n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$

Claim: 3D Chebyshev interpolation is not needed — C can be computed in $\mathcal{O}(d)$ time.

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}),\ t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$

Claim: 3D Chebyshev interpolation is not needed — C can be computed in $\mathcal{O}(d)$ time.

Step 1: Compute the cubic splines $q^{(\ell)}_k$ of data $\mathbf{a}^{(\ell)}_k$ on grid $\{t^{(\ell)}_{i_\ell}\}_{i_\ell=1}^{n_\ell}$.

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}),\ t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$

Claim: 3D Chebyshev interpolation is not needed — $\mathbf{C}$ can be computed in $\mathcal{O}(d)$ time.

Step 1: Compute the cubic splines $q^{(\ell)}_k$ of data $\mathbf{a}^{(\ell)}_k$ on grid $\{t^{(\ell)}_{i_\ell}\}_{i_\ell=1}^{n_\ell}$.

Step 2: Compute 1D Chebyshev interpolant of $q^{(\ell)}_k$: $g^{(\ell)}_k(x) = \sum_{j=1}^{m_\ell} \mathbf{c}^{(\ell)}_k(j) T_{j-1}(x)$

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

ChebTuck approximation: CP tensor input

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}),\ t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$

Claim: 3D Chebyshev interpolation is not needed — $\mathbf{C}$ can be computed in $\mathcal{O}(d)$ time.

Step 1: Compute the cubic splines $q_k^{(\ell)}$ of data $\mathbf{a}_k^{(\ell)}$ on grid $\{t^{(\ell)}_{i_\ell}\}_{i_\ell=1}^{n_\ell}$.

Step 2: Compute 1D Chebyshev interpolant of $q_k^{(\ell)}$: $g_k^{(\ell)}(x) = \sum_{j=1}^{m_\ell} \mathbf{c}_k^{(\ell)}(j) T_{j-1}(x)$

Step 3: $\mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

ChebTuck approximation: CP tensor input

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}),\ t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$

Claim: 3D Chebyshev interpolation is not needed — $\mathbf{C}$ can be computed in $\mathcal{O}(d)$ time.

Step 1: Compute the cubic splines $q_k^{(\ell)}$ of data $\mathbf{a}_k^{(\ell)}$ on grid $\{t_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{n_\ell}$.

Step 2: Compute 1D Chebyshev interpolant of $q_k^{(\ell)}$: $g_k^{(\ell)}(x) = \sum_{j=1}^{m_\ell} \mathbf{c}_k^{(\ell)}(j) T_{j-1}(x)$

Step 3: $\mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$

- CP to Tucker transformation: RHOSVD [Khoromskij/Khoromskaia'09] of $\mathbf{C} \rightsquigarrow$ ChebTuck format.

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

ChebTuck approximation: CP tensor input

- In many applications, f not given explicitly in the full domain $[-1,1]^3$
- But we only have access to its function values at the uniform grid points:

$$\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}),\ t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell,\ i_\ell = 1, \cdots, n_\ell.$$

- We consider here $\mathbf{F}$ is already in CP tensor format[1]: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$

Claim: 3D Chebyshev interpolation is not needed — $\mathbf{C}$ can be computed in $\mathcal{O}(d)$ time.

Step 1: Compute the cubic splines $q^{(\ell)}_k$ of data $\mathbf{a}^{(\ell)}_k$ on grid $\{t^{(\ell)}_{i_\ell}\}_{i_\ell=1}^{n_\ell}$.

Step 2: Compute 1D Chebyshev interpolant of $q^{(\ell)}_k$: $g^{(\ell)}_k(x) = \sum_{j=1}^{m_\ell} \mathbf{c}^{(\ell)}_k(j) T_{j-1}(x)$

Step 3: $\mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}^{(1)}_k \otimes \mathbf{c}^{(2)}_k \otimes \mathbf{c}^{(3)}_k$

- CP to Tucker transformation: RHOSVD [Khoromskij/Khoromskaia'09] of $\mathbf{C} \rightsquigarrow$ ChebTuck format.
- Approx. error can be bounded by $\delta := \max_{k,\ell} \|q^{(\ell)}_k - g^{(\ell)}_k\|_\infty$ & $\varepsilon :=$ RHOSVD truncation error.

[1] for general case, see Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696

ChebTuck approximation error bound: CP tensor input

Theorem (Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696)

$$\max_{i_1,i_2,i_3} \left| \hat{f}_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}) - \mathbf{F}_{i_1,i_2,i_3} \right| \le \|\xi\|_1 e d\delta + \|\xi\| m^{d/2}\varepsilon$$

Remark

- $\delta = \max_{k,\ell} \|q_k^{(\ell)} - g_k^{(\ell)}\|_\infty$, ε is the RHOSVD truncation error.
- $\delta \lesssim \mathrm{e}^{-C_1 m}$ for smooth $q_k^{(\ell)}$
- $\varepsilon \lesssim \mathrm{e}^{-C_2 r}$ for a class of Green's functions [Hackbusch/Khoromskij'08].

Organization

Multi-particle potential

- Calculation of a weighted sum of interaction potential:

$$P(x) = \sum_{v=1}^{N} z_v p(\|x - x_v\|), \quad z_v \in \mathbb{R} \text{ and } x_v, x \in [-1, 1]^d.$$

[2] Bertoglio & Khoromskij, Comp. Phy. Comm. 183.4 (2012): 904-912.

Multi-particle potential

- Calculation of a weighted sum of interaction potential:

$$P(x) = \sum_{v=1}^{N} z_v p(\|x - x_v\|), \quad z_v \in \mathbb{R} \text{ and } x_v, x \in [-1, 1]^d.$$

- $p(\|x\|)$: generating kernel, radial symmetric, slowly decaying, singular at $x = 0$.

[2] Bertoglio & Khoromskij, Comp. Phy. Comm. 183.4 (2012): 904-912.

Multi-particle potential

- Calculation of a weighted sum of interaction potential:

$$P(x) = \sum_{v=1}^{N} z_v p(\|x - x_v\|), \quad z_v \in \mathbb{R} \text{ and } x_v, x \in [-1,1]^d.$$

- $p(\|x\|)$: generating kernel, radial symmetric, slowly decaying, singular at $x = 0$.
- E.g.: Newton (Coulomb) $1/\|x\|$, Slater $\mathrm{e}^{-\lambda\|x\|}$, and Yukawa $\mathrm{e}^{-\lambda\|x\|}/\|x\|$ potentials.

[2] Bertoglio & Khoromskij, Comp. Phy. Comm. 183.4 (2012): 904-912.

Multi-particle potential

- Calculation of a weighted sum of interaction potential:

$$P(x) = \sum_{v=1}^{N} z_v p(\|x - x_v\|), \quad z_v \in \mathbb{R} \text{ and } x_v, x \in [-1, 1]^d.$$

- $p(\|x\|)$: generating kernel, radial symmetric, slowly decaying, singular at $x = 0$.
- E.g.: Newton (Coulomb) $1/\|x\|$, Slater $\mathrm{e}^{-\lambda\|x\|}$, and Yukawa $\mathrm{e}^{-\lambda\|x\|}/\|x\|$ potentials.
- Fact: there exists analytic CP approximation for $p(\|x\|) \rightsquigarrow$ also for $P(x)$[2].
- Focus on Newton.

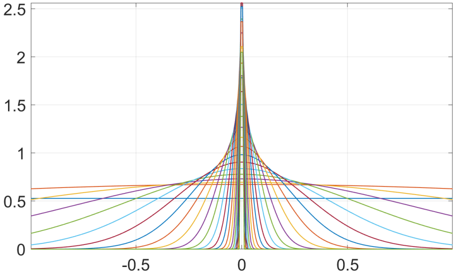
[2] Bertoglio & Khoromskij, Comp. Phy. Comm. 183.4 (2012): 904-912.

Analytic CP approximation for Newton kernel

- $\mathbf{P} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$: projection-collocation tensor of Newton kernel $p(x) = 1/\|x\|$ on a $n_1 \times n_2 \times n_3$ 3D uniform grid:

$$p_{\mathbf{i}} := \int_{[-1,1]^3} \frac{\psi_{\mathbf{i}}(x)}{\|x\|} \,\mathrm{d}\, x = \frac{1}{h^3} \int_{t^{(1)}_{i_1-1}}^{t^{(1)}_{i_1}} \int_{t^{(2)}_{i_2-1}}^{t^{(2)}_{i_2}} \int_{t^{(3)}_{i_3-1}}^{t^{(3)}_{i_3}} \frac{1}{\|x\|} \,\mathrm{d}\, x_1 \,\mathrm{d}\, x_2 \,\mathrm{d}\, x_3,$$

- $\mathbf{P} \approx \mathbf{P}_R = \sum_{k=1}^{R} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}$. Plot of the canonical vectors $\mathbf{p}_k^{(1)}$ along the x-axis:



ChebTuck approximation of the Newton kernel

- Algebraic tensor format of Newton:$\mathbf{P} \approx \mathbf{P}_R = \sum_{k=1}^{R} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}$. Storage: $\mathcal{O}(dnR)$
- ChebTuck storage: $\mathcal{O}(dmr + r^3)$

$$\mathrm{err} := \max_{i_1,i_2} \left| \mathbf{P}_R(i_1, i_2, n/2) - \hat{f}_{\mathbf{m}}(t_{i_1}, t_{i_2}, t_{n/2}) \right| / \max_{i_1,i_2} |\mathbf{P}_R(i_1, i_2, n/2)|$$

n \ m	129	257	513	1025	2049	4097	8193	16385
256	0.41	0.07	$7.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$	$8.5 \cdot 10^{-6}$	$1.6 \cdot 10^{-6}$	$3.5 \cdot 10^{-7}$
512	0.71	0.41	0.07	$7.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$	$8.5 \cdot 10^{-6}$	$1.6 \cdot 10^{-6}$
1024	0.92	0.71	0.41	0.07	$7.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$	$8.5 \cdot 10^{-6}$
2048	1.06	0.92	0.71	0.41	0.07	$7.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$
4096	1.16	1.06	0.92	0.71	0.41	0.07	$7.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-4}$

Polynomial approximation does not work for singular functions! $\rightsquigarrow$ range separation needed.

ChebTuck approximation of the long-range part of Newton kernel

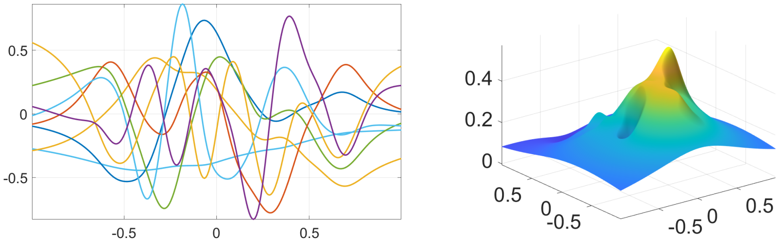
- Algebraic tensor format of Newton: $\mathbf{P} \approx \mathbf{P}_R = \sum_{k=1}^{R} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}$.
- Range-separation $\mathbf{P}_R = \mathbf{P}_{R_s} + \mathbf{P}_{R_l}$[3].
- $\mathbf{P}_{R_s}$: highly localized and sparse.
- $\mathbf{P}_{R_l}$: well approximated by ChebTuck.

n \ m	129	257	513	1025	2049	4097
256	$8.4 \cdot 10^{-8}$	$7.4 \cdot 10^{-8}$	$7.3 \cdot 10^{-8}$	$4.9 \cdot 10^{-9}$	$1.6 \cdot 10^{-9}$	$1.8 \cdot 10^{-10}$
512	$5.2 \cdot 10^{-8}$	$5.2 \cdot 10^{-8}$	$4.6 \cdot 10^{-8}$	$4.6 \cdot 10^{-8}$	$3.0 \cdot 10^{-9}$	$9.9 \cdot 10^{-10}$
1024	$9.4 \cdot 10^{-9}$	$9.4 \cdot 10^{-9}$	$9.4 \cdot 10^{-9}$	$8.7 \cdot 10^{-9}$	$8.6 \cdot 10^{-9}$	$5.7 \cdot 10^{-10}$
2048	$3.6 \cdot 10^{-9}$	$1.7 \cdot 10^{-9}$	$1.7 \cdot 10^{-9}$	$1.7 \cdot 10^{-9}$	$1.6 \cdot 10^{-9}$	$1.6 \cdot 10^{-9}$
4096	$1.1 \cdot 10^{-4}$	$7.5 \cdot 10^{-10}$	$7.4 \cdot 10^{-10}$	$7.4 \cdot 10^{-10}$	$7.4 \cdot 10^{-10}$	$7.1 \cdot 10^{-10}$

Good approximation accuracy with $m \ll n$!

[3] Benner, Khoromskaia, Khoromskij, SISC 40.2 (2018): A1034-A1062.

- $P(x) = \sum_{v=1}^{N} z_v p(\|x - x_v\|)$ is sum of shifted Newton.
- CP format of Newton: $\mathbf{P}_R = \sum_{k=1}^{R} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}$.
- Let $\mathbf{P}_0$ be the projection-collocation tensor of $P(x)$.
- It can be shown: $\mathbf{P}_0$ can be constructed by shifted single Newton $\mathbf{P}_R$[4].
- Similar range-separation: $\mathbf{P}_0 = \mathbf{P}_s + \mathbf{P}_l$.
- For a protein-like molecule with $N = 500$. Plot of canonical vectors of $\mathbf{P}_l$ and $\mathbf{P}_l(:,:,n/2)$.



[4] Khoromskaia, Khoromskij, Comp. Phy. Comm. 185 (2014): 3162-3174.

ChebTuck approximation error for a protein-like molecule

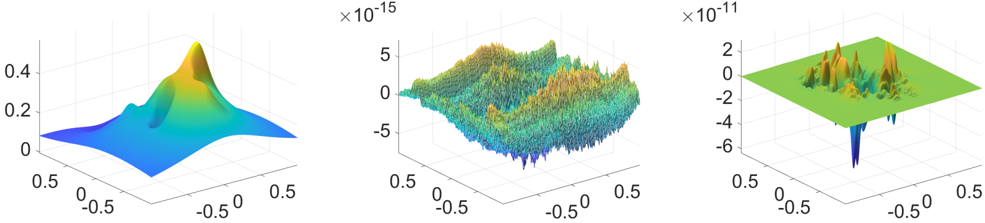


Figure: Left: middle slice $\hat{f}_{\mathbf{m}}(:,:,t_{n/2})$ of ChebTuck format. Middle: error during RHOSVD compression, i.e. $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \tilde{f}_{\mathbf{m}}(:,:,t_{n/2})$. Right: total error $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \mathbf{P}_l(:,:,n/2)$. $n = 2048$, $m = 129$.

Theorem (Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696)

$$\max_{i_1,i_2,i_3} \left| \hat{f}_{\mathbf{m}}(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}) - \mathbf{F}_{i_1,i_2,i_3} \right| \varepsilon \lesssim \|\xi\|_1 e d e^{-C_1 m} + \underbrace{\|\xi\| m^{d/2} \mathrm{e}^{-C_2 r}}_{\textit{Error during RHOSVD approximation}}$$

ChebTuck approximation error for lattice-type structure

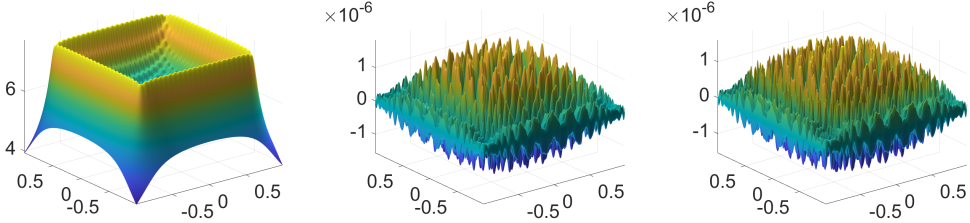


Figure: Left: the middle slice $\hat{f}_{\mathbf{m}}(:,:,t_{n/2})$ of the ChebTuck format. Middle: the error during RHOSVD compression, i.e. $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \tilde{f}_{\mathbf{m}}(:,:,t_{n/2})$. Right: the total error $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \mathbf{P}_l(:,:,n/2)$.

Theorem (Benner, Khoromskaia, Khoromskij, S., 2025, arXiv 2503.01696)

$$\max_{i_1,i_2,i_3} \left| \hat{f}_{\mathbf{m}}(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}) - \mathbf{F}_{i_1,i_2,i_3} \right| \varepsilon \lesssim \|\xi\|_1 e d e^{-C_1 m} + \underbrace{\|\xi\| m^{d/2} \mathrm{e}^{-C_2 r}}_{\textit{Error during RHOSVD approximation}}$$

Conclusions

- A mesh-free ChebTuck format

Conclusions

- A mesh-free ChebTuck format
- Numerical schemes for computing ChebTuck format, with different input assumptions

Conclusions

- A mesh-free ChebTuck format
- Numerical schemes for computing ChebTuck format, with different input assumptions
- In case of rank-structured tensor approximation of target function discretized on large spacial grid: ChebTuck is cheap to construct, accurate and storage efficient

Conclusions

- A mesh-free ChebTuck format
- Numerical schemes for computing ChebTuck format, with different input assumptions
- In case of rank-structured tensor approximation of target function discretized on large spacial grid: ChebTuck is cheap to construct, accurate and storage efficient
- For more details: *A mesh-free hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling*, https://arxiv.org/abs/2503.01696

Conclusions

- A mesh-free ChebTuck format
- Numerical schemes for computing ChebTuck format, with different input assumptions
- In case of rank-structured tensor approximation of target function discretized on large spacial grid: ChebTuck is cheap to construct, accurate and storage efficient
- For more details: *A mesh-free hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling*, https://arxiv.org/abs/2503.01696
- See also: Hashemi and Trefethen, 2017. *Chebfun in three dimensions*.

Conclusions

- A mesh-free ChebTuck format
- Numerical schemes for computing ChebTuck format, with different input assumptions
- In case of rank-structured tensor approximation of target function discretized on large spacial grid: ChebTuck is cheap to construct, accurate and storage efficient
- For more details: *A mesh-free hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling*, https://arxiv.org/abs/2503.01696
- See also: Hashemi and Trefethen, 2017. *Chebfun in three dimensions*.
- See: Strössner, S., and Kressner, 2024. *Approximation in the extended functional tensor train format* for an extension to higher dimensions.

Thank You for Your Attention!