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)

MESIGA25, March 11, 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

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,

$$\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)} \Big(\text{ or Tucker } \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)} \text{ or TT} \Big).$$

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,

$$\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)} \left(\text{ or Tucker } \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)} \text{ or TT} \right).$$

- 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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,

$$\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)} (\text{ or Tucker } \approx \textstyle\sum_{i_1, i_2, i_3 = 1}^{r_1, r_2, r_3} \beta_{i_1, i_2, i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)} \text{ or TT}).$$

- Storage: $\mathcal{O}(d \cdot n \cdot \mathrm{poly}(r))$. Note: full storage $\mathcal{O}(n^d)$ is not necessary.

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,

$$\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)} \left(\text{ or Tucker } \approx \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \beta_{i_1,i_2,i_3} \mathbf{a}_{i_1}^{(1)} \otimes \mathbf{a}_{i_2}^{(2)} \otimes \mathbf{a}_{i_3}^{(3)} \text{ or TT} \right).$$

- Storage: $\mathcal{O}(d \cdot n \cdot \mathrm{poly}(r))$. Note: full storage $\mathcal{O}(n^d)$ is not necessary.
- Disadvantage: large n is required to achieve high accuracy.
- **Mesh-free methods**: functional low-rank tensor approximation.

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$. Storage: $\mathcal{O}(dnr)$.
- **Mesh-free methods**: functional low-rank tensor approximation.

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$. Storage: $\mathcal{O}(dnr)$.
- **Mesh-free methods**: functional low-rank tensor approximation.
- Simplest: Functional CP format (separation of variables):

$$f(x_1, x_2, x_3) \approx \sum_{k=1}^{r} \xi_k g_k^{(1)}(x_1) g_k^{(2)}(x_2) g_k^{(3)}(x_3),$$

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$. Storage: $\mathcal{O}(dnr)$.
- **Mesh-free methods**: functional low-rank tensor approximation.
- Simplest: Functional CP format (separation of variables):

$$f(x_1, x_2, x_3) \approx \sum_{k=1}^{r} \xi_k g_k^{(1)}(x_1) g_k^{(2)}(x_2) g_k^{(3)}(x_3),$$

- where $g_k^{(\ell)}$ are univariate functions parameterized by Chebyshev polynomials

$$g_k^{(\ell)}(x_\ell) = \sum_{j=0}^{m-1} \alpha_{k,j}^{(\ell)} T_j(x_\ell), \; T_j(x) = \cos(j \arccos(x)).$$

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$. Storage: $\mathcal{O}(dnr)$.
- **Mesh-free methods**: functional low-rank tensor approximation.
- Simplest: Functional CP format (separation of variables):

$$f(x_1, x_2, x_3) \approx \sum_{k=1}^{r} \xi_k g_k^{(1)}(x_1) g_k^{(2)}(x_2) g_k^{(3)}(x_3),$$

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

$$\int_{[-1,1]^3} f(x_1, x_2, x_3) \,\mathrm{d}\, x_1 \,\mathrm{d}\, x_2 \,\mathrm{d}\, x_3 \approx \sum_{k=1}^{r} \xi_k \int_{-1}^{1} g_k^{(1)}(x_1) \,\mathrm{d}\, x_1 \int_{-1}^{1} g_k^{(2)}(x_2) \,\mathrm{d}\, x_2 \int_{-1}^{1} g_k^{(3)}(x_3) \,\mathrm{d}\, x_3.$$

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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$. Storage: $\mathcal{O}(dnr)$.
- **Mesh-free methods**: functional low-rank tensor approximation.
- Simplest: Functional CP format (separation of variables):

$$f(x_1, x_2, x_3) \approx \sum_{k=1}^{r} \xi_k g_k^{(1)}(x_1) g_k^{(2)}(x_2) g_k^{(3)}(x_3),$$

- Storage: $\mathcal{O}(dmr)$. $m \ll n$ 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 $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{F} \approx \sum_{k=1}^{r} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$. Storage: $\mathcal{O}(dnr)$.
- **Mesh-free methods**: functional low-rank tensor approximation.
- Simplest: Functional CP format (separation of variables):

$$f(x_1, x_2, x_3) \approx \sum_{k=1}^{r} \xi_k g_k^{(1)}(x_1) g_k^{(2)}(x_2) g_k^{(3)}(x_3),$$

- Storage: $\mathcal{O}(dmr)$. $m \ll n$ for the same accuracy as grid-based methods.
- Often beneficial to impose additional low-rank structures $\rightsquigarrow$ Functional Tucker format.
- This work focuses on the **Functional Tucker format**.

Functional Tucker approximation

- 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.
- **Mesh-free methods**: functional low-rank tensor approximation.

Functional Tucker approximation

- 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.
- **Mesh-free methods**: functional low-rank tensor approximation.
- This work focuses on the **Functional Tucker format**:

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

where $\boldsymbol{\beta} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor and $v_{i_\ell}^{(\ell)} : [-1, 1] \to \mathbb{R}$ are factor functions.

Functional Tucker approximation

- 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.
- **Mesh-free methods**: functional low-rank tensor approximation.
- This work focuses on the **Functional Tucker format**:

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

  where $\boldsymbol{\beta} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor and $v_{i_\ell}^{(\ell)} : [-1,1] \to \mathbb{R}$ are factor functions.
- In case of $v_{i_\ell}^{(\ell)}$ parameterized by Chebyshev polynomials, we call it **ChebTuck format**:

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

Functional Tucker approximation

- 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.
- **Mesh-free methods**: functional low-rank tensor approximation.
- This work focuses on the **Functional Tucker format**:

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

  where $\boldsymbol{\beta} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is the core tensor and $v_{i_\ell}^{(\ell)} : [-1,1] \to \mathbb{R}$ are factor functions.
- In case of $v_{i_\ell}^{(\ell)}$ parameterized by Chebyshev polynomials, we call it **ChebTuck format**:

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

- Goal: given f, compute its ChebTuck approximation.

1. ChebTuck format introduction

2. Numerical schemes for ChebTuck approximation

3. Applications to multi-particle modelling

ChebTuck approximation: functional case

- Consider $f : [-1, 1]^3 \to \mathbb{R}$. 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

- Consider $f : [-1,1]^3 \to \mathbb{R}$. 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

- Consider $f : [-1,1]^3 \to \mathbb{R}$. 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

- Consider $f : [-1,1]^3 \to \mathbb{R}$. 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 $\mathbf{C}$: $\mathbf{C} \approx \hat{\mathbf{C}} = \boldsymbol{\beta} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$

ChebTuck approximation: functional case

- Consider $f : [-1,1]^3 \to \mathbb{R}$. 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

Algorithm 1 ChebTuck Approximation: functional case

Require: Given a function $f : [-1,1]^3 \to \mathbb{R}$, the Chebyshev degrees $\mathbf{m} = (m_1, m_2, m_3)$ and Tucker-ALS truncation error $\varepsilon > 0$.

Ensure: The ChebTuck format $\hat{f}_{\mathbf{m}}$ of f.

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

2: Compute the Chebyshev coefficient tensor $\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)}$

3: Apply Tucker-ALS to $\mathbf{C}$ with tolerance ε to obtain $\boldsymbol{\beta}$ and $V^{(1)}, V^{(2)}, V^{(3)}$

4: Form the factor functions $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$ for $\ell = 1,2,3$.

5: Return $\hat{f}_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \sum_{i_3=1}^{r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3)$.

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

ChebTuck approximation: algebraic case

- 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:

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

- 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:

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

- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.

ChebTuck approximation: algebraic case

- 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:

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

- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- **Extrapolate** from $\mathbf{F}$ to $\mathbf{T}$, using some interpolation method, e.g. multivariate cubic spline interpolation.

ChebTuck approximation: algebraic case

- 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:

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

- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- **Extrapolate** from $\mathbf{F}$ to $\mathbf{T}$, using some interpolation method, e.g. multivariate cubic spline interpolation.
- **Evaluate** the interpolant Q at Chebyshev nodes to obtain $\mathbf{T}$.

ChebTuck approximation: algebraic case

- 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:

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

- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- **Extrapolate** from $\mathbf{F}$ to $\mathbf{T}$, using some interpolation method, e.g. multivariate cubic spline interpolation.
- **Evaluate** the interpolant Q at Chebyshev nodes to obtain $\mathbf{T}$.
- We consider here $\mathbf{F}$ is already in CP tensor format: $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$
  $\rightsquigarrow$ for general case, see arXiv 2503.01696.

ChebTuck approximation: algebraic case (CP tensor input)

- For $\mathbf{F}_{i_1,i_2,i_3} = f(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}),\ t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h_\ell$, $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$.
- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.

ChebTuck approximation: algebraic case (CP tensor input)

- For $\mathbf{F}_{i_1,i_2,i_3} = f(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}),\ t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h_\ell$, $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}_k^{(1)} \otimes \mathbf{a}_k^{(2)} \otimes \mathbf{a}_k^{(3)}$.
- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- Goal: leverage the low-rank structure of $\mathbf{F}$ to **extrapolate** to Chebyshev nodes.
- Idea: construct *univariate* cubic spline interpolations of the canonical vectors $\mathbf{a}_k^{(\ell)}$ in each mode and make tensor product summations of them.

ChebTuck approximation: algebraic case (CP tensor input)

- For $\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3})$, $t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell$, $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$.
- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- Goal: leverage the low-rank structure of $\mathbf{F}$ to **extrapolate** to Chebyshev nodes.
- Idea: construct *univariate* cubic spline interpolations of the canonical vectors $\mathbf{a}^{(\ell)}_k$ in each mode and make tensor product summations of them.
- Let $q^{(\ell)}_k(x_\ell)$ be the cubic spline of the data $\mathbf{a}^{(\ell)}_k$ on the uniform grid $\{t^{(\ell)}_{i_\ell}\}^{n_\ell}_{i_\ell=1}$, i.e., $q^{(\ell)}_k(t^{(\ell)}_{i_\ell}) = \mathbf{a}^{(\ell)}_k(i_\ell)$.

ChebTuck approximation: algebraic case (CP tensor input)

- For $\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3})$, $t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell$, $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$.
- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- Goal: leverage the low-rank structure of $\mathbf{F}$ to **extrapolate** to Chebyshev nodes.
- Idea: construct *univariate* cubic spline interpolations of the canonical vectors $\mathbf{a}^{(\ell)}_k$ in each mode and make tensor product summations of them.
- Let $q^{(\ell)}_k(x_\ell)$ be the cubic spline of the data $\mathbf{a}^{(\ell)}_k$ on the uniform grid $\{t^{(\ell)}_{i_\ell}\}_{i_\ell=1}^{n_\ell}$, i.e., $q^{(\ell)}_k(t^{(\ell)}_{i_\ell}) = \mathbf{a}^{(\ell)}_k(i_\ell)$.
- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q^{(1)}_k(x_1) q^{(2)}_k(x_2) q^{(3)}_k(x_3)$$

ChebTuck approximation: algebraic case (CP tensor input)

- For $\mathbf{F}_{i_1,i_2,i_3} = f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3})$, $t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell$, $\mathbf{F} = \sum_{k=1}^{R} \xi_k \mathbf{a}^{(1)}_k \otimes \mathbf{a}^{(2)}_k \otimes \mathbf{a}^{(3)}_k$.
- First step in our algorithm is to get $\mathbf{T}$, i.e., function values at Chebyshev nodes.
- Goal: leverage the low-rank structure of $\mathbf{F}$ to **extrapolate** to Chebyshev nodes.
- Idea: construct *univariate* cubic spline interpolations of the canonical vectors $\mathbf{a}^{(\ell)}_k$ in each mode and make tensor product summations of them.
- Let $q^{(\ell)}_k(x_\ell)$ be the cubic spline of the data $\mathbf{a}^{(\ell)}_k$ on the uniform grid $\{t^{(\ell)}_{i_\ell}\}_{i_\ell=1}^{n_\ell}$, i.e., $q^{(\ell)}_k(t^{(\ell)}_{i_\ell}) = \mathbf{a}^{(\ell)}_k(i_\ell)$.
- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q^{(1)}_k(x_1) q^{(2)}_k(x_2) q^{(3)}_k(x_3)$$

- Evaluate Q at Chebyshev nodes to obtain $\mathbf{T} \rightsquigarrow \mathbf{T}$ itself is in CP format

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.
- Let $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ be the evaluation of $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.
- Let $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ be the evaluation of $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.
- $\rightsquigarrow \mathbf{T} = \sum_{k=1}^{R} \xi_k \mathbf{q}_k^{(1)} \otimes \mathbf{q}_k^{(2)} \otimes \mathbf{q}_k^{(3)}$.

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.
- Let $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ be the evaluation of $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.
- $\rightsquigarrow \mathbf{T} = \sum_{k=1}^{R} \xi_k \mathbf{q}_k^{(1)} \otimes \mathbf{q}_k^{(2)} \otimes \mathbf{q}_k^{(3)}$.
- $\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} = \sum_{k=1}^{R} \xi_k \left(W^{(1)}\mathbf{q}_k^{(1)}\right) \otimes \left(W^{(2)}\mathbf{q}_k^{(2)}\right) \otimes \left(W^{(3)}\mathbf{q}_k^{(3)}\right)$.

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.
- Let $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ be the evaluation of $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.
- $\rightsquigarrow \mathbf{T} = \sum_{k=1}^{R} \xi_k \mathbf{q}_k^{(1)} \otimes \mathbf{q}_k^{(2)} \otimes \mathbf{q}_k^{(3)}$.
- $\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} = \sum_{k=1}^{R} \xi_k \left(W^{(1)}\mathbf{q}_k^{(1)}\right) \otimes \left(W^{(2)}\mathbf{q}_k^{(2)}\right) \otimes \left(W^{(3)}\mathbf{q}_k^{(3)}\right)$.
- Let $\mathbf{c}_k^{(\ell)} = W^{(\ell)}\mathbf{q}_k^{(\ell)}$, $\rightsquigarrow \mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$.

Compute T and Chebyshev coefficients C

- Construct tensor product spline $Q(x_1, x_2, x_3)$ by

$$Q(x_1, x_2, x_3) = \sum_{k=1}^{R} \xi_k q_k^{(1)}(x_1) q_k^{(2)}(x_2) q_k^{(3)}(x_3)$$

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.
- Let $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ be the evaluation of $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.
- $\rightsquigarrow \mathbf{T} = \sum_{k=1}^{R} \xi_k \mathbf{q}_k^{(1)} \otimes \mathbf{q}_k^{(2)} \otimes \mathbf{q}_k^{(3)}$.
- $\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} = \sum_{k=1}^{R} \xi_k \left(W^{(1)} \mathbf{q}_k^{(1)}\right) \otimes \left(W^{(2)} \mathbf{q}_k^{(2)}\right) \otimes \left(W^{(3)} \mathbf{q}_k^{(3)}\right)$.
- Let $\mathbf{c}_k^{(\ell)} = W^{(\ell)} \mathbf{q}_k^{(\ell)}$, $\rightsquigarrow \mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$.
- Note: $\mathbf{c}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ are the Chebyshev coefficients of the univariate spline $q_k^{(\ell)}$, i.e., $g_k^{(\ell)}(x) = \sum_{j=1}^{m_\ell} \mathbf{c}_k^{(\ell)}(j) T_{j-1}(x) \approx$ and interpolate $q_k^{(\ell)}(x)$.

- $\mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$.
- CP to Tucker transformation: RHOSVD of $\mathbf{C}$.
- Let $C^{(\ell)} = [\mathbf{c}_1^{(\ell)}, \cdots, \mathbf{c}_R^{(\ell)}] \in \mathbb{R}^{m_\ell \times R}$, $\boldsymbol{\xi} \in \mathbb{R}^{R \times R \times R}$ diagonal tensor $\rightsquigarrow$ $\mathbf{C} = \boldsymbol{\xi} \times_1 C^{(1)} \times_2 C^{(2)} \times_3 C^{(3)}$.
- Compute rank-r_ℓ approximation of $C^{(\ell)} \approx V^{(\ell)} U^{(\ell)}$ $\rightsquigarrow$ $\mathbf{C} \approx \underbrace{(\boldsymbol{\xi} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)})}_{=:\boldsymbol{\beta} \in \mathbb{R}^{r_1 \times r_2 \times r_3}} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)}$.

Algorithm 2 ChebTuck Approximation: algebraic case (CP tensor input, Reduced)

Require: Given a tensor $\mathbf{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ in CP format discretizing function $f : [-1, 1]^3 \to \mathbb{R}$ on the uniform grid, the degree of Chebyshev polynomials in each dimension m_1, m_2, m_3 and Tucker-ALS truncation error $\varepsilon > 0$.

Ensure: The ChebTuck format $\hat{f}_{\mathbf{m}}$ of f.

1: Construct univariate splines $q_k^{(\ell)}(x_\ell)$ of data $\mathbf{a}_k^{(\ell)}$.

2: Evaluate $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.

3: Compute the Chebyshev coefficients $\mathbf{c}_k^{(\ell)} = W^{(\ell)} \mathbf{q}_k^{(\ell)}$.

4: Apply RHOSVD to the CP tensor $\mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$ to obtain its Tucker format $\boldsymbol{\beta}$ and $V^{(1)}, V^{(2)}, V^{(3)}$.

5: Perform lines 4-5 in Algorithm 1.

ChebTuck approximation error bound: CP tensor input, Reduced

- Note: $\mathbf{T}$ itself is in CP format: $\mathbf{T}_{i_1,i_2,i_3} = \sum_{k=1}^{R} \xi_k q_k^{(1)}(s_{i_1}^{(1)}) q_k^{(2)}(s_{i_2}^{(2)}) q_k^{(3)}(s_{i_3}^{(3)})$.
- Let $\mathbf{q}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ be the evaluation of $q_k^{(\ell)}(x)$ at $\{s_{i_\ell}^{(\ell)}\}_{i_\ell=1}^{m_\ell}$, i.e., $\mathbf{q}_k^{(\ell)}(i_\ell) = q_k^{(\ell)}(s_{i_\ell}^{(\ell)})$.
- $\mathbf{C} = \mathbf{T} \times_1 W^{(1)} \times_2 W^{(2)} \times_3 W^{(3)} = \sum_{k=1}^{R} \xi_k \left(W^{(1)}\mathbf{q}_k^{(1)}\right) \otimes \left(W^{(2)}\mathbf{q}_k^{(2)}\right) \otimes \left(W^{(3)}\mathbf{q}_k^{(3)}\right)$.
- Let $\mathbf{c}_k^{(\ell)} = W^{(\ell)}\mathbf{q}_k^{(\ell)}$, $\rightsquigarrow \mathbf{C} = \sum_{k=1}^{R} \xi_k \mathbf{c}_k^{(1)} \otimes \mathbf{c}_k^{(2)} \otimes \mathbf{c}_k^{(3)}$.
- Note: $\mathbf{c}_k^{(\ell)} \in \mathbb{R}^{m_\ell}$ are the Chebyshev coefficients of the univariate spline $q_k^{(\ell)}$, i.e., $g_k^{(\ell)}(x) = \sum_{j=1}^{m_\ell} \mathbf{c}_k^{(\ell)}(j) T_{j-1}(x) \approx$ and interpolate $q_k^{(\ell)}(x)$.
- Let $\delta = \max_{k,\ell} \|q_k^{(\ell)} - g_k^{(\ell)}\|_\infty$, ε be the RHOSVD truncation error.

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

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

Organization

Multi-particle potential

- Calculation of a weighted sum of interaction potential:

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

[1] 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 and singular at the origin $x = 0$.

[1] 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 and singular at the origin $x = 0$.
- E.g.: Newton (Coulomb) $1/\|x\|$, Slater $\mathrm{e}^{-\lambda\|x\|}$, and Yukawa $\mathrm{e}^{-\lambda\|x\|}/\|x\|$ potentials.

[1]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 and singular at the origin $x = 0$.
- 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 $P(x)$[1].
- Focus on Newton.

[1] 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:

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

- Multi-index: $\mathbf{i} = (i_1, i_2, i_3) \in \{1, \cdots, n\}^3$; grid points: $t^{(\ell)}_{i_\ell} = -1 + (i_\ell - 1)h_\ell$; grid size: $h_\ell = 2/(n_\ell - 1)$
- Basis function $\psi_{\mathbf{i}}$: piecewise constant functions $\psi_{\mathbf{i}}(x) = \psi_{i_1}(x_1)\psi_{i_2}(x_2)\psi_{i_3}(x_3)$, $\psi_{i_\ell}(x_\ell) = \chi_{[t^{(\ell)}_{i_\ell-1}, t^{(\ell)}_{i_\ell})}(x_\ell)/h_\ell$
- WLOG, assume $n_1 = n_2 = n_3 = n$ and use h and t_{i_ℓ} to denote grid size and points.
- $\exists$ quadrature points and weights $\{\eta_k, \omega_k\}_{k=1}^M$ s.t. $\left| \frac{1}{\|x\|} - \sum_{k=1}^M \omega_k \mathrm{e}^{-\eta_k^2 \|x\|^2} \right| \lesssim \mathrm{e}^{-\alpha\sqrt{M}}$
- Replace $1/\|x\|$ by the above exponential sum:

$$p_{\mathbf{i}} \approx \frac{1}{h^3} \sum_{k=1}^{M} \omega_k \left(\int_{t_{i_1-1}}^{t_{i_1}} \mathrm{e}^{-\eta_k^2 x_1^2} \,\mathrm{d}\, x_1 \right) \left(\int_{t_{i_2-1}}^{t_{i_2}} \mathrm{e}^{-\eta_k^2 x_2^2} \,\mathrm{d}\, x_2 \right) \left(\int_{t_{i_3-1}}^{t_{i_3}} \mathrm{e}^{-\eta_k^2 x_3^2} \,\mathrm{d}\, x_3 \right)$$

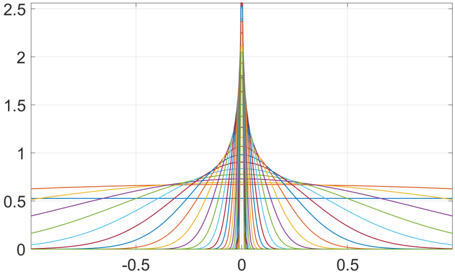
$$\rightsquigarrow \mathbf{P} \approx \mathbf{P}_R = \sum_{k=1}^{R} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}$$

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: $\hat{f}_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1,i_2,i_3=1}^{r_1,r_2,r_3} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3)$, $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m_\ell} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell)$. 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	$7.7 \cdot 10^{-3}$	$6.9 \cdot 10^{-4}$

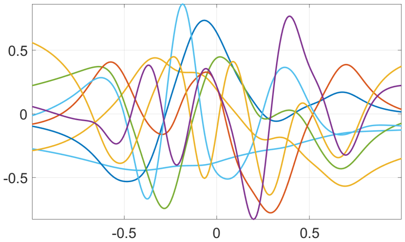
- Algebraic tensor format of Newton: $\mathbf{P} \approx \mathbf{P}_R = \sum_{k=1}^{R} \mathbf{p}_k^{(1)} \otimes \mathbf{p}_k^{(2)} \otimes \mathbf{p}_k^{(3)}$.
- Range-separation $\mathbf{P}_R = \mathbf{P}_{R_s} + \mathbf{P}_{R_l}$[2].
- $\mathbf{P}_{R_s}$: highly localized and sparse.
- $\mathbf{P}_{R_l}$: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

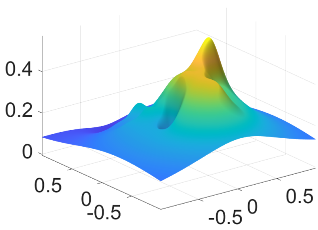
n \ m	129	257	513	1025	2049	4097
256	$8.4 \cdot 10^{-8}$	$7.4 \cdot 10^{-8}$	$7.3 \cdot 10^{-8}$	$4.9 \cdot 10^{-9}$	$1.6 \cdot 10^{-9}$	$1.8 \cdot 10^{-10}$
512	$5.2 \cdot 10^{-8}$	$5.2 \cdot 10^{-8}$	$4.6 \cdot 10^{-8}$	$4.6 \cdot 10^{-8}$	$3.0 \cdot 10^{-9}$	$9.9 \cdot 10^{-10}$
1024	$9.4 \cdot 10^{-9}$	$9.4 \cdot 10^{-9}$	$9.4 \cdot 10^{-9}$	$8.7 \cdot 10^{-9}$	$8.6 \cdot 10^{-9}$	$5.7 \cdot 10^{-10}$
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 $m \ll n$!

[2] 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 $\mathbf{P}_R$[3].
- 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 $\mathbf{P}_l(:,:,n/2)$.





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

ChebTuck approximation error for a protein-like molecule

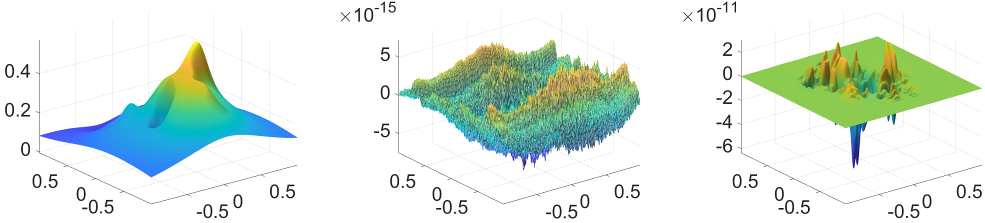


Figure: Left: the middle slice $\hat{f}_{\mathbf{m}}(:,:,t_{n/2})$ of the ChebTuck format. Middle: the error during RHOSVD compression, i.e. $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \tilde{f}_{\mathbf{m}}(:,:,t_{n/2})$. Right: the total error $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \mathbf{P}_l(:,:,n/2)$.

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

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

ChebTuck approximation error for lattice-type structure

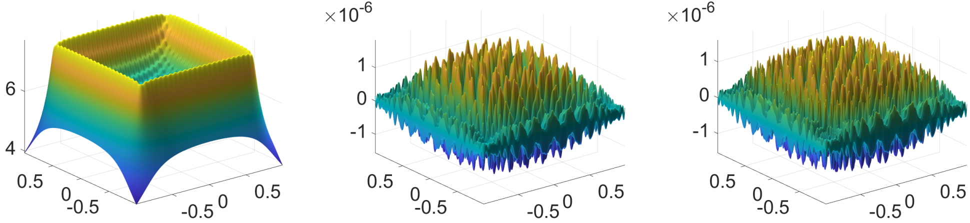


Figure: Left: the middle slice $\hat{f}_{\mathbf{m}}(:,:,t_{n/2})$ of the ChebTuck format. Middle: the error during RHOSVD compression, i.e. $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \tilde{f}_{\mathbf{m}}(:,:,t_{n/2})$. Right: the total error $\hat{f}_{\mathbf{m}}(:,:,t_{n/2}) - \mathbf{P}_l(:,:,n/2)$.

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

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

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
- Future directions: more application scenarios, extend to higher dimensions, ...
- For more details: *A mesh-free hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling*, https://arxiv.org/abs/2503.01696

Thank You for Your Attention!