MAX PLANCK GESELLSCHAFT

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

Bridging continuous and discrete tensor representations of multivariate functions via QTT

Bonan Sun (Max Planck Institute Magdeburg)

METT XI, Jan 9, 2026

Based on a joint work with
Peter Benner (MPI Magdeburg)
Boris Khoromskij (MPI Magdeburg and MPI MiS Leipzig)

Partners:

MAX PLANCK INSTITUTE FOR MATHEMATICS IN THE SCIENCES

Organization

1. Introduction
2. QTT approximation of polynomials
3. Fully discrete format
4. Numerical experiments
5. Conclusion

Discrete and continuous tensor formats

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

1. **Grid-based methods**: **discrete Tucker format** of function related tensor $\mathbf{F}$ (contains, e.g., function values on a grid):

$$\mathbf{F}(i_1,i_2,i_3) \approx \sum_{j_1=1}^{R}\sum_{j_2=1}^{R}\sum_{j_3=1}^{R} \boldsymbol{\beta}_{j_1,j_2,j_3}\mathbf{u}_{j_1}^{(1)}(i_1)\mathbf{u}_{j_2}^{(2)}(i_2)\mathbf{u}_{j_3}^{(3)}(i_3), \mathbf{u}_{j_\ell}^{(\ell)} \in \mathbb{R}^n,\ \mathbf{F} \in \mathbb{R}^{n\times n\times n}.$$

2. **Mesh-free methods**: **functional Tucker format** of f:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) := \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1)v_{i_2}^{(2)}(x_2)v_{i_3}^{(3)}(x_3) \tag{1}$$

Discrete and continuous tensor formats

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

1. **Grid-based methods**: **discrete Tucker format** of function related tensor $\mathbf{F}$ (contains, e.g., function values on a grid):

$$\mathbf{F}(i_1, i_2, i_3) \approx \sum_{j_1=1}^{R} \sum_{j_2=1}^{R} \sum_{j_3=1}^{R} \boldsymbol{\beta}_{j_1,j_2,j_3} \mathbf{u}_{j_1}^{(1)}(i_1) \mathbf{u}_{j_2}^{(2)}(i_2) \mathbf{u}_{j_3}^{(3)}(i_3), \mathbf{u}_{j_\ell}^{(\ell)} \in \mathbb{R}^n, \ \mathbf{F} \in \mathbb{R}^{n\times n\times n}.$$

2. **Mesh-free methods**: **functional Tucker format** of f:

$$\begin{aligned} f(x_1,x_2,x_3) &\approx f_{\mathbf{m}}(x_1,x_2,x_3) := \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3) \\ v_{i_\ell}^{(\ell)}(x_\ell) &= \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell), \ V^{(\ell)} \in \mathbb{R}^{m\times R}, \ T_{j_\ell}(x) = \cos(j_\ell \arccos(x)). \end{aligned} \tag{1}$$

We call Eq. (1) the **Chebyshev-Tucker (ChebTuck) format**.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h, \quad h = 2/(n-1).$$

- **Motivation**:
  - Efficient application of discrete operators (differentiation, integration, convolution).
  - Offline-online workflows: expensive offline construction of continuous surrogate, fast online evaluation.
  - High-resolution discretizations for accurate simulations.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.

- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h, \quad h = 2/(n-1).$$

- A **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.

- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- **Goal**: Transform ChebTuck format to a discrete tensor format on a fine grid $n \gg m$:

$$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t_{i_1}^{(1)}, t_{i_2}^{(2)}, t_{i_3}^{(3)}), \quad t_{i_\ell}^{(\ell)} = -1 + (i_\ell - 1)h, \quad h = 2/(n-1).$$

- A **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.
- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].

ChebTuck to Grid-based tensor

- Consider $f : [-1, 1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1, x_2, x_3) \approx f_{\mathbf{m}}(x_1, x_2, x_3) = \sum_{i_1=1}^{R} \sum_{i_2=1}^{R} \sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.
- A **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.
- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].
- Known: QTT ranks of polynomials p of degree m are bounded by $m+1$ [Khoromskij '11, Oseledets '13] (numerically even $\log m$) $\rightsquigarrow \mathcal{O}(DRn)$ reduced to $\mathcal{O}(DRm^2 \log n)$ (numerically $\mathcal{O}(DR \log^2 m \log n)$).

ChebTuck to Grid-based tensor

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

  with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DRm + R^D)$.

- A **naive approach** yields a discrete Tucker:

$$\mathbf{F}_{\mathbf{m}} = \boldsymbol{\beta} \times_1 U_1 \times_2 U_2 \times_3 U_3, \quad U_\ell(i_\ell, j_\ell) = v_{j_\ell}^{(\ell)}(t_{i_\ell}^{(\ell)})$$

  i.e., each column of U_ℓ contains the discretization of a Chebyshev poly. $v_{j_\ell}^{(\ell)}(x_\ell)$ on the grid $t_{i_\ell}^{(\ell)}$.

- **Storage**: $\mathcal{O}(DRn + R^D)$. **Additional storage** compared to ChebTuck: $\mathcal{O}(DRn)$, $n \gg m$.
- **Remedy:** store columns of U_ℓ as quantized tensor trains (QTT) [Khoromskij '11].
- Known: QTT ranks of polynomials p of degree m are bounded by $m+1$ [Khoromskij '11, Oseledets '13] (numerically even $\log m$) $\rightsquigarrow \mathcal{O}(DRn)$ reduced to $\mathcal{O}(DRm^2 \log n)$ (numerically $\mathcal{O}(DR \log^2 m \log n)$).
- **Fundamental task**: approximate a polynomial p in QTT format efficiently.

From ChebTuck to QTT-Tucker

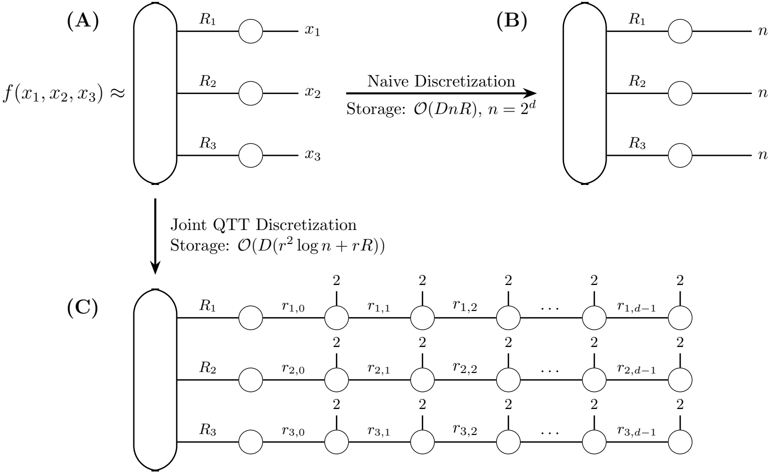


Figure: (A) Continuous ChebTuck. (B) Discrete Tucker ($\mathcal{O}(n)$ storage). (C) QTT-Tucker-like ($\mathcal{O}(\log n)$ storage).

Organization

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.
- Using the multi-index mapping $(i_1, \cdots, i_d) \mapsto i_{\leq d} = 1 + (i_1 - 1) + 2 \cdot (i_2 - 1) + \cdots + 2^{d-1} \cdot (i_d - 1)$ for $i_\ell = 1, 2$, we can reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\mathbf{P}(i_1, \cdots, i_d) = p(x_{i_{\leq d}}) = p\Big(-1 + h/2 + (i_1 - 1)h + 2 \cdot (i_2 - 1)h + \cdots + 2^{d-1} \cdot (i_d - 1)h\Big)$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.
- Using the multi-index mapping $(i_1, \cdots, i_d) \mapsto i_{\leq d} = 1 + (i_1 - 1) + 2 \cdot (i_2 - 1) + \cdots + 2^{d-1} \cdot (i_d - 1)$ for $i_\ell = 1, 2$, we can reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\mathbf{P}(i_1, \ldots, i_d) = p(x_{i_{\leq d}}) = p\Big(\underbrace{-1 + h/2 + (i_1 - 1)h}_{=:x_{i_1}^{(1)}} + \underbrace{2 \cdot (i_2 - 1)h}_{=:x_{i_2}^{(2)}} + \cdots + \underbrace{2^{d-1} \cdot (i_d - 1)h}_{=:x_{i_d}^{(d)}}\Big)$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.
- Using the multi-index mapping $(i_1, \cdots, i_d) \mapsto i_{\leq d} = 1 + (i_1 - 1) + 2 \cdot (i_2 - 1) + \cdots + 2^{d-1} \cdot (i_d - 1)$ for $i_\ell = 1, 2$, we can reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\begin{aligned}\mathbf{P}(i_1, \ldots, i_d) = p(x_{i_{\leq d}}) &= p\Big(\underbrace{-1 + h/2 + (i_1 - 1)h}_{=:x_{i_1}^{(1)}} + \underbrace{2 \cdot (i_2 - 1)h}_{=:x_{i_2}^{(2)}} + \cdots + \underbrace{2^{d-1} \cdot (i_d - 1)h}_{=:x_{i_d}^{(d)}}\Big) \\ &= p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).\end{aligned}$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^{n}$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^{n} \in \mathbb{R}^{2^d}$.
- Using the multi-index mapping $(i_1, \cdots, i_d) \mapsto i_{\le d} = 1 + (i_1 - 1) + 2 \cdot (i_2 - 1) + \cdots + 2^{d-1} \cdot (i_d - 1)$ for $i_\ell = 1, 2$, we can reshape $\mathbf{p}$ to a d-dimensional tensor $\mathbf{P} \in \mathbb{R}^{2 \times \cdots \times 2}$:

$$\begin{aligned}\mathbf{P}(i_1, \ldots, i_d) = p(x_{i_{\le d}}) &= p\Big(\underbrace{-1 + h/2 + (i_1 - 1)h}_{=: x_{i_1}^{(1)}} + \underbrace{2 \cdot (i_2 - 1)h}_{=: x_{i_2}^{(2)}} + \cdots + \underbrace{2^{d-1} \cdot (i_d - 1)h}_{=: x_{i_d}^{(d)}} \Big) \\ &= p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).\end{aligned}$$

- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) \cdots \mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.
- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) \cdots \mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Computational methods for the TT of P

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets '11].
2. **Constructive method**: there exists an analytic formula[1] for the cores $\mathbf{G}_\ell(i_\ell) \in \mathbb{R}^{(m+1)\times(m+1)}$.

[1] Oseledets, Constructive Approximation 37(1):1–18 (2013)

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.
- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Computational methods for the TT of P

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets '11].
2. **Constructive method**: there exists an analytic formula[1] for the cores $\mathbf{G}_\ell(i_\ell) \in \mathbb{R}^{(m+1)\times(m+1)}$.

Disadvantages: Method 1 does not exploit the 1D nature of the problem, just views $\mathbf{P}$ as a black box tensor. Method 2 quickly becomes numerically unstable & produces pessimistic TT ranks $m+1$.

[1] Oseledets, Constructive Approximation 37(1):1–18 (2013)

Approximation of a polynomial in QTT format

- For a polynomial $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ in $[-1, 1]$, discretizing it on uniform grid $\{x_i := -1 + h/2 + (i-1)h\}_{i=1}^n$ with $h = 2/n$ and $n = 2^d$ yields $\mathbf{p} = \{p(x_i)\}_{i=1}^n \in \mathbb{R}^{2^d}$.
- QTT of p is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Computational methods for the TT of P

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets '11].
2. **Constructive method**: there exists an analytic formula[1] for the cores $\mathbf{G}_\ell(i_\ell) \in \mathbb{R}^{(m+1)\times(m+1)}$.

Disadvantages: Method 1 does not exploit the 1D nature of the problem, just views $\mathbf{P}$ as a black box tensor. Method 2 quickly becomes numerically unstable & produces pessimistic TT ranks $m+1$.

3. **Our novel method**: Constructive (thus faster than Method 1), stable and rank adaptive.

[1] Oseledets, Constructive Approximation 37(1):1–18 (2013)

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta} C_{\alpha+\beta}^{\alpha}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \dots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \dots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta)x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta}C_{\alpha+\beta}^{\alpha}$ for $\alpha+\beta \le m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\le k}(x) := [1, x, \dots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\le m}(x)^\top M X_{\le m}(y)$.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta)x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta}C_{\alpha+\beta}^{\alpha}$ for $\alpha+\beta \le m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\le k}(x) := [1, x, \ldots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\le m}(x)^\top M X_{\le m}(y)$.
3. Let $x = x_{i_1}^{(1)}$ and $y = x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}$, then separate $x_{i_1}^{(1)}$ from the rest:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \underbrace{X_{\le m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1)\in\mathbb{R}^{1\times(m+1)}} X_{\le m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$$

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha, \beta) x^\alpha y^\beta$ with $M(\alpha, \beta) = p_{\alpha+\beta} C_{\alpha+\beta}^{\alpha}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\leq k}(x) := [1, x, \ldots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\leq m}(x)^\top M X_{\leq m}(y)$.
3. Let $x = x_{i_1}^{(1)}$ and $y = x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}$, then separate $x_{i_1}^{(1)}$ from the rest:
$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \underbrace{X_{\leq m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$$
4. $\mathbf{G}_2(i_2)$ is constructed such that $X_{\leq m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{G}_2(i_2) X_{\leq m}(x_{i_3}^{(3)} + \cdots + x_{i_d}^{(d)})$ holds
$$\Longrightarrow \mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) X_{\leq m}(x_{i_3}^{(3)} + \cdots + x_{i_d}^{(d)}).$$

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta)x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta}C_{\alpha+\beta}^{\alpha}$ for $\alpha+\beta \le m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\le k}(x) := [1, x, \ldots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\le m}(x)^\top M X_{\le m}(y)$.
3. Let $x = x_{i_1}^{(1)}$ and $y = x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}$, then separate $x_{i_1}^{(1)}$ from the rest:
$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \underbrace{X_{\le m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1)\in\mathbb{R}^{1\times(m+1)}} X_{\le m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$$
4. $\mathbf{G}_2(i_2)$ is constructed such that $X_{\le m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{G}_2(i_2)X_{\le m}(x_{i_3}^{(3)} + \cdots + x_{i_d}^{(d)})$ holds
$$\Longrightarrow \mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)X_{\le m}(x_{i_3}^{(3)} + \cdots + x_{i_d}^{(d)}).$$
5. Other cores $\mathbf{G}_3(i_3), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta)x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta}C_{\alpha+\beta}^{\alpha}$ for $\alpha+\beta \le m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\le k}(x) := [1, x, \ldots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\le m}(x)^\top M X_{\le m}(y)$.
3. Let $x = x_{i_1}^{(1)}$ and $y = x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}$, then separate $x_{i_1}^{(1)}$ from the rest:
$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \underbrace{X_{\le m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1)\in\mathbb{R}^{1\times(m+1)}} X_{\le m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$$
4. Other cores $\mathbf{G}_2(i_2), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta} C_{\alpha+\beta}^{\alpha}$ for $\alpha + \beta \leq m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\leq k}(x) := [1, x, \ldots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\leq m}(x)^\top M X_{\leq m}(y)$.
3. Let $x = x_{i_1}^{(1)}$ and $y = x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}$, then separate $x_{i_1}^{(1)}$ from the rest:
$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \underbrace{X_{\leq m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1) \in \mathbb{R}^{1\times(m+1)}} X_{\leq m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$$
4. Other cores $\mathbf{G}_2(i_2), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Disadvantage: Unstable since the cores contain large binomial coefficients & powers e.g. $(x_{i_1}^{(1)})^\alpha$.

Approximation of a polynomial in QTT format

QTT of $p(x) = p_0 + p_1 x + \cdots + p_m x^m$ is defined as a TT of $\mathbf{P}$:

$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{P}(i_1, \ldots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1, \ldots, i_d = 1, 2.$$

Recall Oseledets' constructive method

1. Notice $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha,\beta)x^\alpha y^\beta$ with $M(\alpha,\beta) = p_{\alpha+\beta}C_{\alpha+\beta}^{\alpha}$ for $\alpha+\beta \le m$ otherwise 0, i.e., M is upper anti-triangular.
2. Introduce the notation: $X_{\le k}(x) := [1, x, \ldots, x^k]^\top \in \mathbb{R}^{k+1}$, then $p(x+y) = X_{\le m}(x)^\top M X_{\le m}(y)$.
3. Let $x = x_{i_1}^{(1)}$ and $y = x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}$, then separate $x_{i_1}^{(1)}$ from the rest:
$$p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \underbrace{X_{\le m}(x_{i_1}^{(1)})^\top M}_{=\mathbf{G}_1(i_1)\in\mathbb{R}^{1\times(m+1)}} X_{\le m}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}).$$
4. Other cores $\mathbf{G}_2(i_2), \cdots, \mathbf{G}_d(i_d)$ are constructed similarly.

Disadvantage: Unstable since the cores contain large binomial coefficients & powers e.g. $(x_{i_1}^{(1)})^\alpha$.

Fix: replace x^α, y^β by Chebyshev polynomials: $p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M_m^{\text{cheb}}(\alpha,\beta)T_\alpha(x)T_\beta(y)$.

Direct construction in Chebyshev basis

Fixing idea: Replace monomial expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^{\alpha} y^{\beta} \tag{2}$$

by Chebyshev expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M_m^{\mathrm{cheb}}(\alpha,\beta) T_{\alpha}(x) T_{\beta}(y). \tag{3}$$

Direct construction in Chebyshev basis

Fixing idea: Replace monomial expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta \tag{2}$$

by Chebyshev expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M_m^{\text{cheb}}(\alpha,\beta) T_\alpha(x) T_\beta(y). \tag{3}$$

- **Subtle but crucial difference**: (2) holds for all $x, y \in \mathbb{R}$ for fixed coefficients M, but (3) only holds for $x \in I_x$ and $y \in I_y$ with some intervals I_x, I_y and the coefficients M_m^{cheb} depend accordingly on these intervals.

Direct construction in Chebyshev basis

Fixing idea: Replace monomial expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M(\alpha,\beta) x^\alpha y^\beta \tag{2}$$

by Chebyshev expansion

$$p(x+y) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{m} M_m^{\mathrm{cheb}}(\alpha,\beta) T_\alpha(x) T_\beta(y). \tag{3}$$

- **Subtle but crucial difference**: (2) holds for all $x, y \in \mathbb{R}$ for fixed coefficients M, but (3) only holds for $x \in I_x$ and $y \in I_y$ with some intervals I_x, I_y and the coefficients M_m^{cheb} depend accordingly on these intervals.
- Notation: Scaled Chebyshev polynomials on $I = [a,b]$:

  $T_\ell^I(x) := T_\ell\left((2x-(a+b))/(b-a)\right)$, $T_{\leq k}^I(x) := [1, T_1^I(x), \cdots, T_k^I(x)]^\top \in \mathbb{R}^{k+1}$ for $x \in I = [a,b]$.

Rank-adaptive construction in Chebyshev basis (Forward pass)

Key tool: 1D Chebyshev interpolation

For a polynomial $p : I \to \mathbb{R}$ of degree $\leq m$, its Chebyshev expansion is exact:

$$p(x) = \sum_{\ell=0}^{m} c_\ell T_\ell^I(x) = \mathbf{c}^\top T_{\leq m}^I(x), \quad \text{coefficients } \mathbf{c} = W\mathbf{p}$$

where $\mathbf{p}$ contains evaluations at Chebyshev points and W is the DCT (inverse) matrix.

1.

Rank-adaptive construction in Chebyshev basis (Forward pass)

Key tool: 1D Chebyshev interpolation

For a polynomial $p : I \to \mathbb{R}$ of degree $\leq m$, its Chebyshev expansion is exact:

$$p(x) = \sum_{\ell=0}^{m} c_\ell T_\ell^I(x) = \mathbf{c}^\top T_{\leq m}^I(x), \quad \text{coefficients } \mathbf{c} = W\mathbf{p}$$

where $\mathbf{p}$ contains evaluations at Chebyshev points and W is the DCT (inverse) matrix.

Let $I_{\geq k} := [\sum_{\ell=k}^{d} x_1^{(\ell)}, \sum_{\ell=k}^{d} x_2^{(\ell)}]$, then we have $x_{i_k}^{(k)} + x_{i_{k+1}}^{(k+1)} + \cdots + x_{i_d}^{(d)} \in I_{\geq k}$.

1.

Rank-adaptive construction in Chebyshev basis (Forward pass)

Key tool: 1D Chebyshev interpolation

For a polynomial $p : I \to \mathbb{R}$ of degree $\leq m$, its Chebyshev expansion is exact:

$$p(x) = \sum_{\ell=0}^{m} c_\ell T_\ell^I(x) = \mathbf{c}^\top T_{\leq m}^I(x), \quad \text{coefficients } \mathbf{c} = W\mathbf{p}$$

where $\mathbf{p}$ contains evaluations at Chebyshev points and W is the DCT (inverse) matrix.

Let $I_{\geq k} := [\sum_{\ell=k}^d x_1^{(\ell)}, \sum_{\ell=k}^d x_2^{(\ell)}]$, then we have $x_{i_k}^{(k)} + x_{i_{k+1}}^{(k+1)} + \cdots + x_{i_d}^{(d)} \in I_{\geq k}$.

1. Let $\mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$ be the Chebyshev coefficients of the function $I_{\geq 2} \ni x \mapsto p(x_{i_1}^{(1)} + x)$:

$$p(x_{i_1}^{(1)} + x) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x) \text{ for } x \in I_{\geq 2},\ i_1 = 1, 2,$$

Rank-adaptive construction in Chebyshev basis (Forward pass)

Key tool: 1D Chebyshev interpolation

For a polynomial $p : I \to \mathbb{R}$ of degree $\leq m$, its Chebyshev expansion is exact:

$$p(x) = \sum_{\ell=0}^{m} c_\ell T_\ell^I(x) = \mathbf{c}^\top T_{\leq m}^I(x), \quad \text{coefficients } \mathbf{c} = W\mathbf{p}$$

where $\mathbf{p}$ contains evaluations at Chebyshev points and W is the DCT (inverse) matrix.

Let $I_{\geq k} := [\sum_{\ell=k}^d x_1^{(\ell)}, \sum_{\ell=k}^d x_2^{(\ell)}]$, then we have $x_{i_k}^{(k)} + x_{i_{k+1}}^{(k+1)} + \cdots + x_{i_d}^{(d)} \in I_{\geq k}$.

1. Let $\mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$ be the Chebyshev coefficients of the function $I_{\geq 2} \ni x \mapsto p(x_{i_1}^{(1)} + x)$:

$$p(x_{i_1}^{(1)} + x) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x) \text{ for } x \in I_{\geq 2},\ i_1 = 1, 2, \text{ and in particular}$$

$$\mathbf{P}(i_1, i_2, \cdots, i_d) = p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)})$$

Rank-adaptive construction in Chebyshev basis (Forward pass)

Key tool: 1D Chebyshev interpolation

For a polynomial $p : I \to \mathbb{R}$ of degree $\leq m$, its Chebyshev expansion is exact:

$$p(x) = \sum_{\ell=0}^{m} c_\ell T_\ell^I(x) = \mathbf{c}^\top T_{\leq m}^I(x), \quad \text{coefficients } \mathbf{c} = W\mathbf{p}$$

where $\mathbf{p}$ contains evaluations at Chebyshev points and W is the DCT (inverse) matrix.

Let $I_{\geq k} := [\sum_{\ell=k}^d x_1^{(\ell)}, \sum_{\ell=k}^d x_2^{(\ell)}]$, then we have $x_{i_k}^{(k)} + x_{i_{k+1}}^{(k+1)} + \cdots + x_{i_d}^{(d)} \in I_{\geq k}$.

1. Let $\mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$ be the Chebyshev coefficients of the function $I_{\geq 2} \ni x \mapsto p(x_{i_1}^{(1)} + x)$:

$$p(x_{i_1}^{(1)} + x) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x) \text{ for } x \in I_{\geq 2},\ i_1 = 1, 2, \text{ and in particular}$$

$$\mathbf{P}(i_1, i_2, \cdots, i_d) = p(x_{i_1}^{(1)} + x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)})$$

The first QTT core can already be taken as $\mathbf{G}_1(i_1) = \mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$.

Rank-adaptive construction in Chebyshev basis (Forward pass)

1. For $\mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$ being the Chebyshev coefficients of $I_{\geq 2} \ni x \mapsto p(x_{i_1}^{(1)} + x)$, we have

$$\mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) \tag{4}$$

Rank-adaptive construction in Chebyshev basis (Forward pass)

1. For $\mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$ being the Chebyshev coefficients of $I_{\geq 2} \ni x \mapsto p(x_{i_1}^{(1)} + x)$, we have

$$\mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) \tag{4}$$

2. We compute a LRA (e.g. SVD) of the Chebyshev coefficients:
$$\begin{bmatrix} \mathbf{c}_1^{\text{cheb}} \\ \mathbf{c}_2^{\text{cheb}} \end{bmatrix} \approx U_1 V_1^\top = \begin{bmatrix} U_{1,1} \\ U_{1,2} \end{bmatrix} V_1^\top \text{ with } U_1 \in \mathbb{R}^{2\times r_1}, V_1 \in \mathbb{R}^{(m+1)\times r_1}.$$

Rank-adaptive construction in Chebyshev basis (Forward pass)

1. For $\mathbf{c}_{i_1}^{\text{cheb}} \in \mathbb{R}^{1\times(m+1)}$ being the Chebyshev coefficients of $I_{\geq 2} \ni x \mapsto p(x_{i_1}^{(1)} + x)$, we have
$$\mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) \tag{4}$$
2. We compute a LRA (e.g. SVD) of the Chebyshev coefficients:
$$\begin{bmatrix} \mathbf{c}_1^{\text{cheb}} \\ \mathbf{c}_2^{\text{cheb}} \end{bmatrix} \approx U_1 V_1^\top = \begin{bmatrix} U_{1,1} \\ U_{1,2} \end{bmatrix} V_1^\top \text{ with } U_1 \in \mathbb{R}^{2\times r_1}, V_1 \in \mathbb{R}^{(m+1)\times r_1}.$$
3. Thus $\mathbf{c}_{i_1}^{\text{cheb}} \approx U_{1,i_1} V_1^\top$. Then Eq. (4) becomes
$$\begin{aligned}\mathbf{P}(i_1, i_2, \cdots, i_d) = \mathbf{c}_{i_1}^{\text{cheb}} T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) &\approx U_{1,i_1} V_1^\top T_{\leq m}^{I_{\geq 2}}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)}) \\ &= U_{1,i_1} v_{\leq r_1}^{(2)}(x_{i_2}^{(2)} + \cdots + x_{i_d}^{(d)})\end{aligned}$$
where the polynomial vector $v_j^{(2)}(y)$ on the interval $I_{\geq 2}$ is defined as
$v_{\leq r_1}^{(2)}(y) := [v_1^{(2)}(y), \cdots, v_{r_1}^{(2)}(y)]^\top \in \mathbb{R}^{r_1}$ with $v_j^{(2)}(y) := V_1(:, j)^\top T_{\leq m}^{I_{\geq 2}}(y)$ for $j = 1, \cdots, r_1$,
The first QTT core can thus be taken as $\mathbf{G}_1(i_1) = U_{1,i_1}$.

Rank-adaptive construction in Chebyshev basis (Forward pass)

3. We obtained from the previous step:

$$\mathbf{P}(i_1, i_2, \cdots, i_d) \approx U_{1,i_1} v^{(2)}_{\leq r_1}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}).$$

4. For the 2nd QTT core, each $I_{\geq 3} \ni x \mapsto v^{(2)}_j(x^{(2)}_{i_2} + x)$ for $j = 1, \cdots, r_1$ and $i_2 = 1, 2$ is a polynomial of degree $\leq m \implies$ has $\leq m+1$ non-zero Cheb. coeff. denoted by $C^{\text{cheb}}_{2,i_2}(j, :) \in \mathbb{R}^{m+1}$:

$$v^{(2)}_{\leq r_1}(x^{(2)}_{i_2} + \cdots + x^{(d)}_{i_d}) = C^{\text{cheb}}_{2,i_2} T^{I_{\geq 3}}_{\leq m}(x^{(3)}_{i_3} + \cdots + x^{(d)}_{i_d}). \tag{5}$$

We then compute a LRA of $\begin{bmatrix} C^{\text{cheb}}_{2,1} \\ C^{\text{cheb}}_{2,2} \end{bmatrix} \approx U_2 {V_2}^\top = \begin{bmatrix} U_{2,1} \\ U_{2,2} \end{bmatrix} {V_2}^\top$ with $U_2 \in \mathbb{R}^{2r_1 \times r_2}, V_2 \in \mathbb{R}^{(m+1)\times r_2}$. Thus

$$\mathbf{P}(i_1, i_2, \cdots, i_d) \approx U_{1,i_1} C^{\text{cheb}}_{2,i_2} T^{I_{\geq 3}}_{\leq m}(x^{(3)}_{i_3} + \cdots + x^{(d)}_{i_d}) \approx U_{1,i_1} U_{2,i_2} {V_2}^\top T^{I_{\geq 3}}_{\leq m}(x^{(3)}_{i_3} + \cdots + x^{(d)}_{i_d})$$

The second QTT core is therefore $\mathbf{G}_2(i_2) = U_{2,i_2}$. Other cores are constructed similarly.

Rank-adaptive construction (Error control)

- **Backward pass**: Perform right-to-left SVD sweep to further compress the ranks.
- SVD tolerances in forward and backward passes: $\varepsilon_k^{(f)}$ and $\varepsilon_k^{(b)}$ for $k = 1, \ldots, d-1$.
- We have guaranteed element-wise error control.

Theorem [Benner, Khoromskij, S., in preparation, 2026]

For the QTT approximation of the polynomial $p(x)$ constructed by the rank-adaptive method, we have

$$|\mathbf{P} - \hat{\mathbf{P}}| \leq \sqrt{m+1} \sum_{k=1}^{d-1} \varepsilon_k^{(f)} + \sum_{k=1}^{d-1} \varepsilon_{k+1}^{(b)}$$

Organization

Joint QTT approximation

- Consider $f : [-1,1]^3 \to \mathbb{R}$ approximated in **ChebTuck format**:

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) = \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R} \boldsymbol{\beta}_{i_1,i_2,i_3} v_{i_1}^{(1)}(x_1) v_{i_2}^{(2)}(x_2) v_{i_3}^{(3)}(x_3),$$

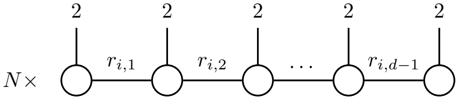
with $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$. **Storage**: $\mathcal{O}(DmR + R^D)$.

- In ChebTuck, we have multiple polynomials $v_{i_\ell}^{(\ell)}(x_\ell)$ for $i_\ell = 1, \ldots, R$ in each dimension ℓ.
- Approximating each separately is inefficient.
- **Joint QTT approximation**:
  - Treat polynomial index as an additional mode.
  - First core $\mathbf{G}_1$ becomes a 3D tensor (or last core).
  - Subsequent cores are shared among all polynomials.

(A)

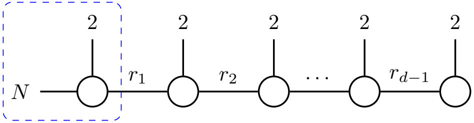
Storage

$\mathcal{O}(dqr^2 \times N)$



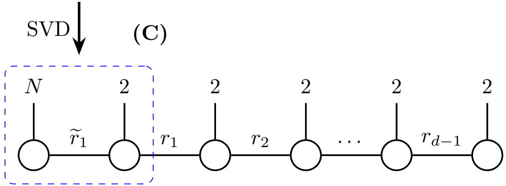
(B)

$\mathcal{O}\big(dq(r_{\text{Joint}}^2 + N r_1)\big)$



(C)

$\mathcal{O}\big(dq(r_{\text{Joint}}^2 + N \widetilde{r}_1)\big)$

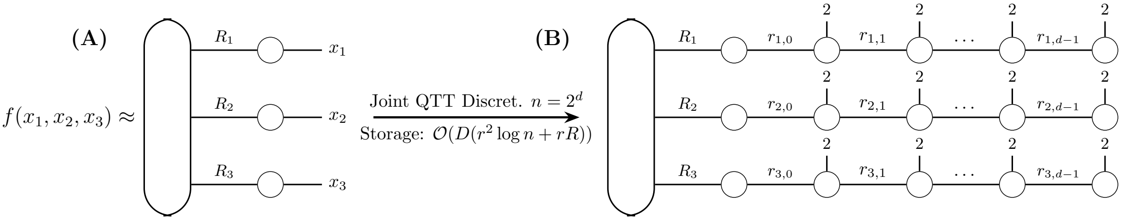


- Combining ChebTuck

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

with Joint QTT:

$$\mathbf{F}(i_1, i_2, i_3) \approx \boldsymbol{\beta} \times_1 \mathbf{U}^{(1)}(i_1) \times_2 \mathbf{U}^{(2)}(i_2) \times_3 \mathbf{U}^{(3)}(i_3).$$



Organization

Random polynomials in monomial basis

- Non-trivial, but practical and not pathological (not too oscillatory) polynomials.
- Polynomials of degree m with random coefficients:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m.$$

- a_k follow the distribution $a_k \sim \mathcal{N}(b\mathrm{e}^{-ak}, c \cdot b\mathrm{e}^{-ak})$ for $k = 1, \ldots, 300$.
- Decay parameters $a \sim U(0, 0.1)$ and $b \sim U(0, 10)$.
- Relative noise level $c = 0.1$.
- Discretization on uniform grid with $n = 2^{20}$ points (20 dimensional QTT tensor).
- Tolerances of all SVDs in our method are set to 10^{-12}.
- `dmrg_cross` tolerance is set to 10^{-12}.

Random polynomials in monomial basis

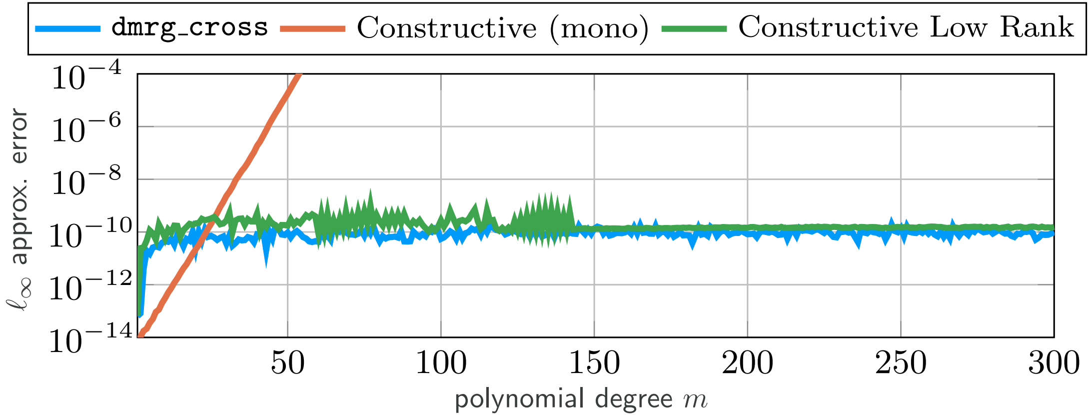


Figure: TTCross vs. Constructive (mono) vs. Constructive Low Rank.

Random polynomials in monomial basis

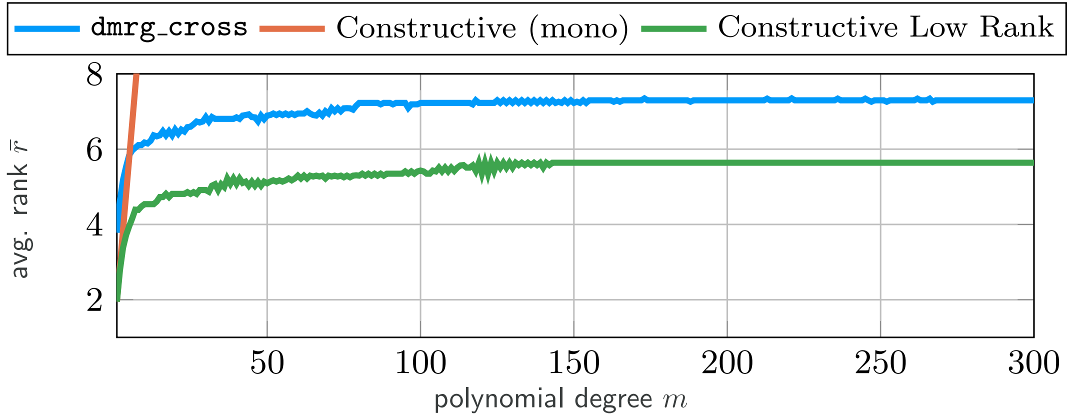


Figure: TTCross vs. Constructive (mono) vs. Constructive Low Rank.

Random polynomials in monomial basis

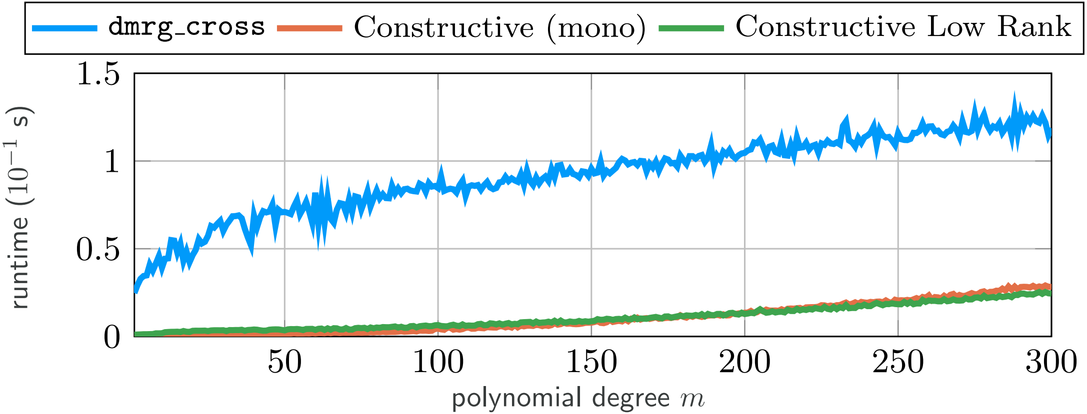


Figure: TTCross vs. Constructive (mono) vs. Constructive Low Rank.

Random polynomials in Chebyshev basis

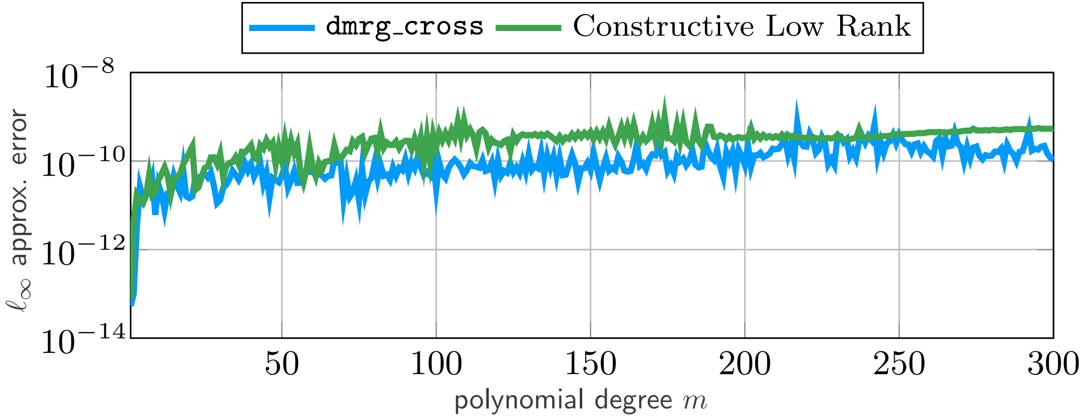


Figure: TTCross vs. Constructive Low Rank.

Random polynomials in Chebyshev basis

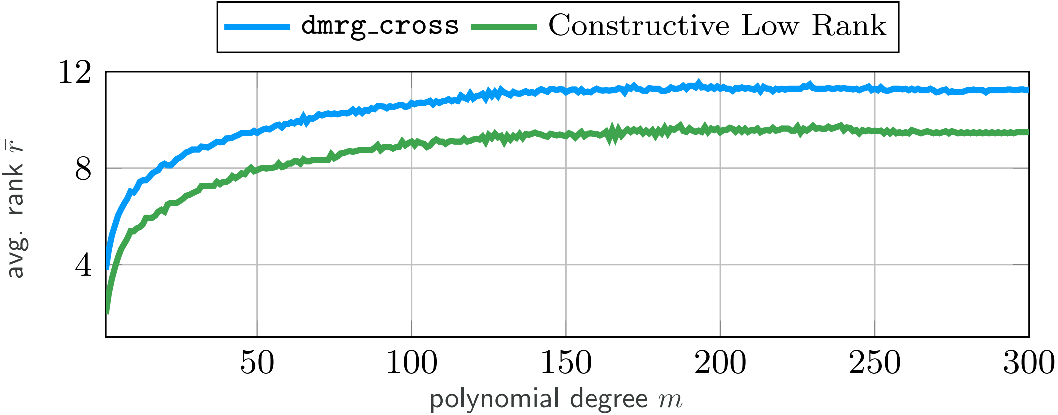


Figure: TTCross vs. Constructive Low Rank.

Random polynomials in Chebyshev basis

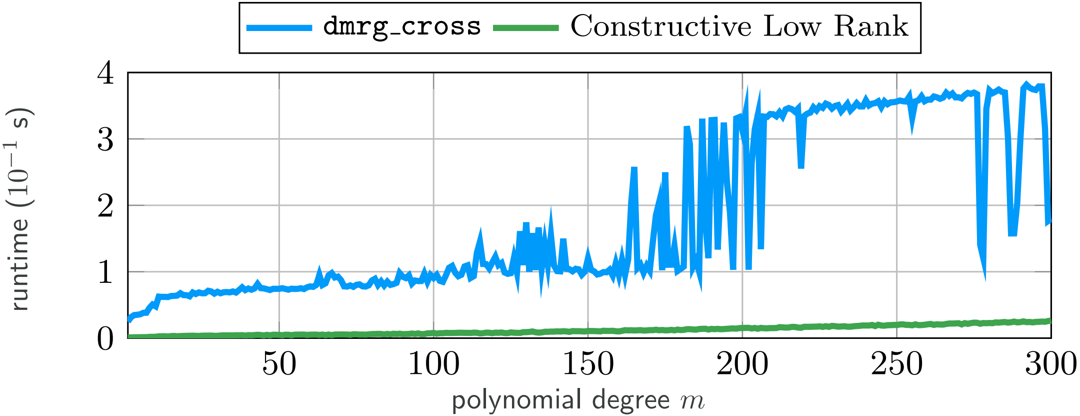


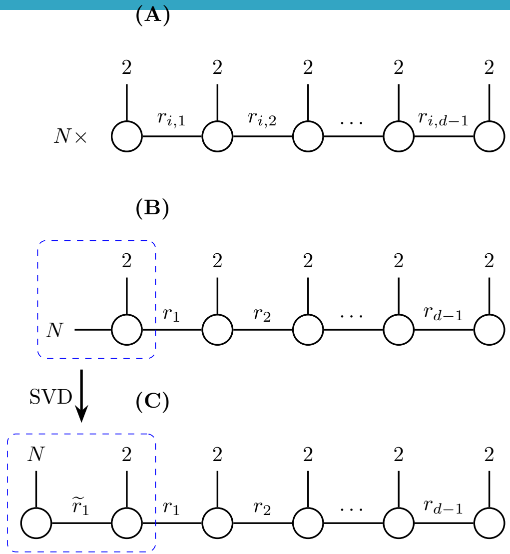
Figure: TTCross vs. Constructive Low Rank.

Storage

$\mathcal{O}(dqr^2 \times N)$

$\mathcal{O}\big(dq(r_{\text{Joint}}^2 + Nr_1)\big)$

$\mathcal{O}\big(dq(r_{\text{Joint}}^2 + N\widetilde{r}_1)\big)$



Test functions

We consider the following 3 test functions $f : [-1,1]^3 \to \mathbb{R}$:

1. **Biomolecule potential** f_1: the multi-particle potential of the protein Fasciculin 1 comprising 1,228 atoms[2].
2. **Runge function** f_2: the classical 3-dimensional Runge function[3] given by

$$f_2(x,y,z) = \frac{1}{1+25(x^2+y^2+z^2)}. \tag{6}$$

3. **Wagon function** f_3: the SIAM 100-Dollar, 100-Digit Challenge function[4] defined by

$$\begin{aligned} f_3(x,y,z) = e^{\sin(50x)} + \sin(60e^y)\sin(60z) + \sin(70\sin(x))\cos(10z) \\ + \sin(\sin(80y)) - \sin(10(x+z)) + \frac{x^2+y^2+z^2}{4}. \end{aligned} \tag{7}$$

[2] M.H. Le Du, P. Marchot, P.E. Bougis, and J.C. Fontecilla-Camps, J. Biol. Chem., 267:22122–30, 1992.
[3] B. Hashemi and L. N. Trefethen, SIAM J. Sci. Comput., 39(5):C341–C363, 2017.
[4] Folkmar Bornemann, Dirk Laurie, Stan Wagon, and Jörg Waldvogel. SIAM, 2004.

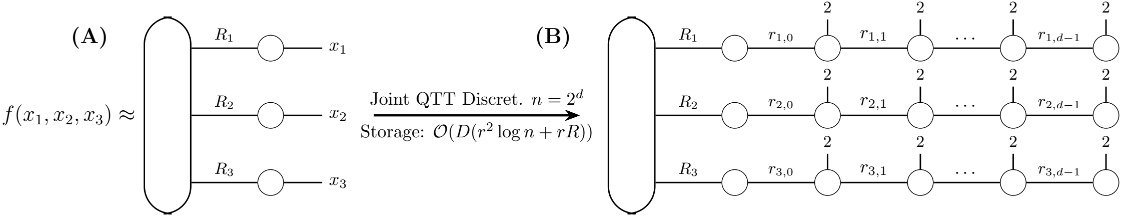
The fully discrete QTT-Tucker-like format

- Combining ChebTuck

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

where $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\; T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$ with Joint QTT:

$$\mathbf{F}(i_1,i_2,i_3) \approx \boldsymbol{\beta} \times_1 \mathbf{U}^{(1)}(i_1) \times_2 \mathbf{U}^{(2)}(i_2) \times_3 \mathbf{U}^{(3)}(i_3).$$



The fully discrete QTT-Tucker-like format

- Combining ChebTuck

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

where $v_{i_\ell}^{(\ell)}(x_\ell) = \sum_{j_\ell=1}^{m} V_{j_\ell,i_\ell}^{(\ell)} T_{j_\ell-1}(x_\ell),\ T_{j_\ell}(x) = \cos(j_\ell \arccos(x))$ with Joint QTT:

$$\mathbf{F}(i_1, i_2, i_3) \approx \boldsymbol{\beta} \times_1 \mathbf{U}^{(1)}(i_1) \times_2 \mathbf{U}^{(2)}(i_2) \times_3 \mathbf{U}^{(3)}(i_3).$$

Function	**Tucker ranks**	**Polynomial degrees**
f_1	(32, 28, 32)	(129, 129, 129)
f_2	(17, 17, 17)	(189, 189, 189)
f_3	(4, 3, 5)	(662, 1052, 129)

Joint QTT approximation of multiple polynomials

function	method	configuration	ℓ_∞ error	storage	avg. rank $\bar{r}$	runtime (10^{-1} s)
f_1	Constructive Low Rank	Separate	2.93×10^{-10}	63,394	7.03	4.1
	`dmrg_cross`	Separate	1.92×10^{-11}	96,414	8.68	29.4
	Constructive Low Rank	**Joint**	$\mathbf{7.88 \times 10^{-11}}$	**17,198**	19.66	**0.2**
	`dmrg_cross`	Joint	7.56	7,824	13.06	4.5
f_2	Constructive Low Rank	Separate	2.92×10^{-10}	39,594	7.48	3.2
	`dmrg_cross`	Separate	1.17×10^{-10}	59,546	9.24	27.0
	Constructive Low Rank	**Joint**	$\mathbf{1.24 \times 10^{-10}}$	**4,240**	9.76	**0.2**
	`dmrg_cross`	Joint	4.45×10^{-8}	5,275	10.93	4.1
f_3	Constructive Low Rank	Separate	2.54×10^{-8}	10,076	7.93	3.9
	`dmrg_cross`	Separate	7.05×10^{-9}	14,620	9.56	7.2
	Constructive Low Rank	**Joint**	$\mathbf{4.99 \times 10^{-9}}$	**3,820**	9.53	**0.9**
	`dmrg_cross`	Joint	3.77×10^{-8}	5,342	11.27	7.5

Table: ChebTuck factors from biomolecule f_1, Runge f_2, and wagon f_3; $n = 2^{20}$.

Fully discrete QTT–Tucker format

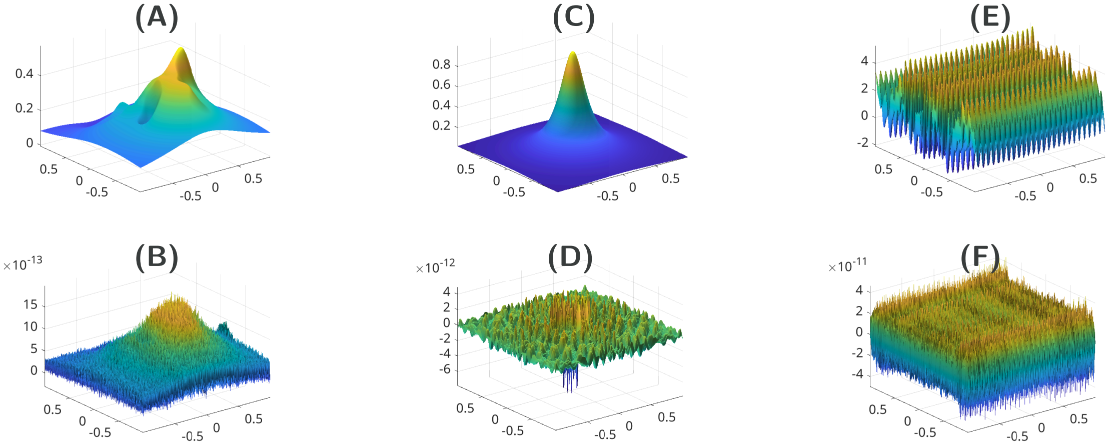


Figure: **(A)**, **(C)**, **(E)**: Surfaces reconstructed from the fully discrete QTT-Tucker format for $f_i(x, y, 0)$.
(B), **(D)**, **(F)**: Corresponding pointwise errors compared to the continuous tensors.

Conclusion

- A framework to bridge continuous (functional) and discrete (grid-based) tensor formats via QTT.
- Developed stable and rank-adaptive algorithms for joint QTT approximation of polynomials.
- Numerical experiments demonstrate stability, efficiency, and storage savings.
- **Outlook**: Extend it to general univariate functions and multivariate ridge functions and more application scenarios.
- For more details and reference:

1. P. Benner, B. Khoromskij, B. Sun. *Bridging continuous and discrete tensor representations of multivariate functions using QTT*, in preparation, 2026.
2. P. Benner, B. Khoromskij, V. Khoromskaia, B. Sun. *A mesh-free hybrid Chebyshev-Tucker tensor format with applications to multi-particle modelling*, arXiv:2505.02319, 2025.

Thank you for your attention!