MAX PLANCK GESELLSCHAFT
MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG
MAX PLANCK INSTITUTE FOR MATHEMATICS IN THE SCIENCES

Bridging continuous and discrete tensor representations of multivariate functions via QTT

Peter Benner[1], Boris N. Khoromskij[1,2], Venera Khoromskaia[2], Bonan Sun[1]

[1]Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany (benner@mpi-magdeburg.mpg.de, bsun@mpi-magdeburg.mpg.de)
[2]Max Planck Institute for Mathematics in the Sciences, 04103 Leipzig, Germany (bokh@mis.mpg.de, vekh@mis.mpg.de)

Chebyshev-Tucker (ChebTuck) format

Discrete and continuous tensor formats

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

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

$$\mathbf{F} \approx \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R}\beta_{i_1,i_2,i_3}\mathbf{a}^{(1)}_{i_1}\otimes\mathbf{a}^{(2)}_{i_2}\otimes\mathbf{a}^{(3)}_{i_3} \in \mathbb{R}^{n\times n\times n},$$
$$\mathbf{F}_{i_1,i_2,i_3} := f(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}),\ t^{(\ell)}_{i_\ell} = -1+(i_\ell-1)h,\ h=2/(n-1),\ i_\ell = 1,\cdots,n. \quad (1)$$

- Storage: $\mathcal{O}(DRn+R^D)$. Disadvantage: *large* n required to achieve high accuracy.

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

$$f(x_1,x_2,x_3) \approx f_{\mathbf{m}}(x_1,x_2,x_3) := \sum_{i_1=1}^{R}\sum_{i_2=1}^{R}\sum_{i_3=1}^{R}\beta_{i_1,i_2,i_3}v^{(1)}_{i_1}(x_1)v^{(2)}_{i_2}(x_2)v^{(3)}_{i_3}(x_3)$$
$$v^{(\ell)}_{i_\ell}(x_\ell) = \sum_{j_\ell=1}^{m} V^{(\ell)}_{j_\ell,i_\ell}T_{j_\ell-1}(x_\ell),\ V^{(\ell)}\in\mathbb{R}^{m\times R},\ T_{j}(x)=\cos(j_\ell \arccos(x)). \quad (2)$$

- Storage: $\mathcal{O}(DRm+R^D)$. $m \ll n$ for the same accuracy as grid-based methods.
- We call it **ChebTuck format**. It is also the format Chebfun3 [3] assumes.

Three natural tasks	Input	Output	Note
Function to ChebTuck (See [1,3])	$f:[-1,1]^3\to\mathbb{R}$	ChebTuck $f_{\mathbf{m}}$	f can be evaluated freely in $[-1,1]^3$
Grid to ChebTuck (See [1])	grid-based $\mathbf{F}$ of f	ChebTuck $f_{\mathbf{m}}$	Preferably $m \ll n$
ChebTuck to Grid (our focus)	ChebTuck $f_{\mathbf{m}}$ of f	grid-based $\mathbf{F}$	$n \gg m$, but storage largely reduced by QTT

ChebTuck to Grid-based tensor

- Given ChebTuck $f_{\mathbf{m}}$ (2) of f, to get a grid-based tensor, naive approach yields a discrete Tucker:
$\mathbf{F}_{\mathbf{m}}(i_1,i_2,i_3) := f_{\mathbf{m}}(t^{(1)}_{i_1}, t^{(2)}_{i_2}, t^{(3)}_{i_3}) \Rightarrow \mathbf{F}_{\mathbf{m}} = \beta\times_1 U_1\times_2 U_2\times_3 U_3,\ U_\ell(i_\ell,j_\ell) = v^{(\ell)}_{j_\ell}(t^{(\ell)}_{i_\ell})$ (c.f. (1)),
i.e., each column of U_ℓ contains the discretization of a polynomial $v^{(\ell)}_{j_\ell}(x_\ell)$ on grid $\{t^{(\ell)}_{i_\ell}\}$.
- **Additional storage**: $\mathcal{O}(DRn)$. **Remedy**: store columns of U_ℓ as QTT [Khoromskij'11].

Definition (Quantized Tensor Trains (QTT) format of a univariate function)

Let $p:[-1,1]\to\mathbb{R}$ be a univariate function. Discretizing it on a uniform grid $\{x_i := -1+(i-1)h\}_{i=1}^n$ with $h=2/(n-1)$, $n=2^d$ yields $\mathbf{p}=[p(x_i)]_{i=1}^n\in\mathbb{R}^n$.
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 $i_\ell=1,2$ reshapes $\mathbf{p}$ into a d-dimensional tensor $\mathbf{P}\in\mathbb{R}^{2\times\cdots\times 2}$:

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

QTT of p is defined as a TT (with cores $\mathbf{G}_\ell(i_\ell)\in\mathbb{R}^{r_{\ell-1}\times r_\ell}$ and ranks $r_\ell\le r$) of $\mathbf{P}$:

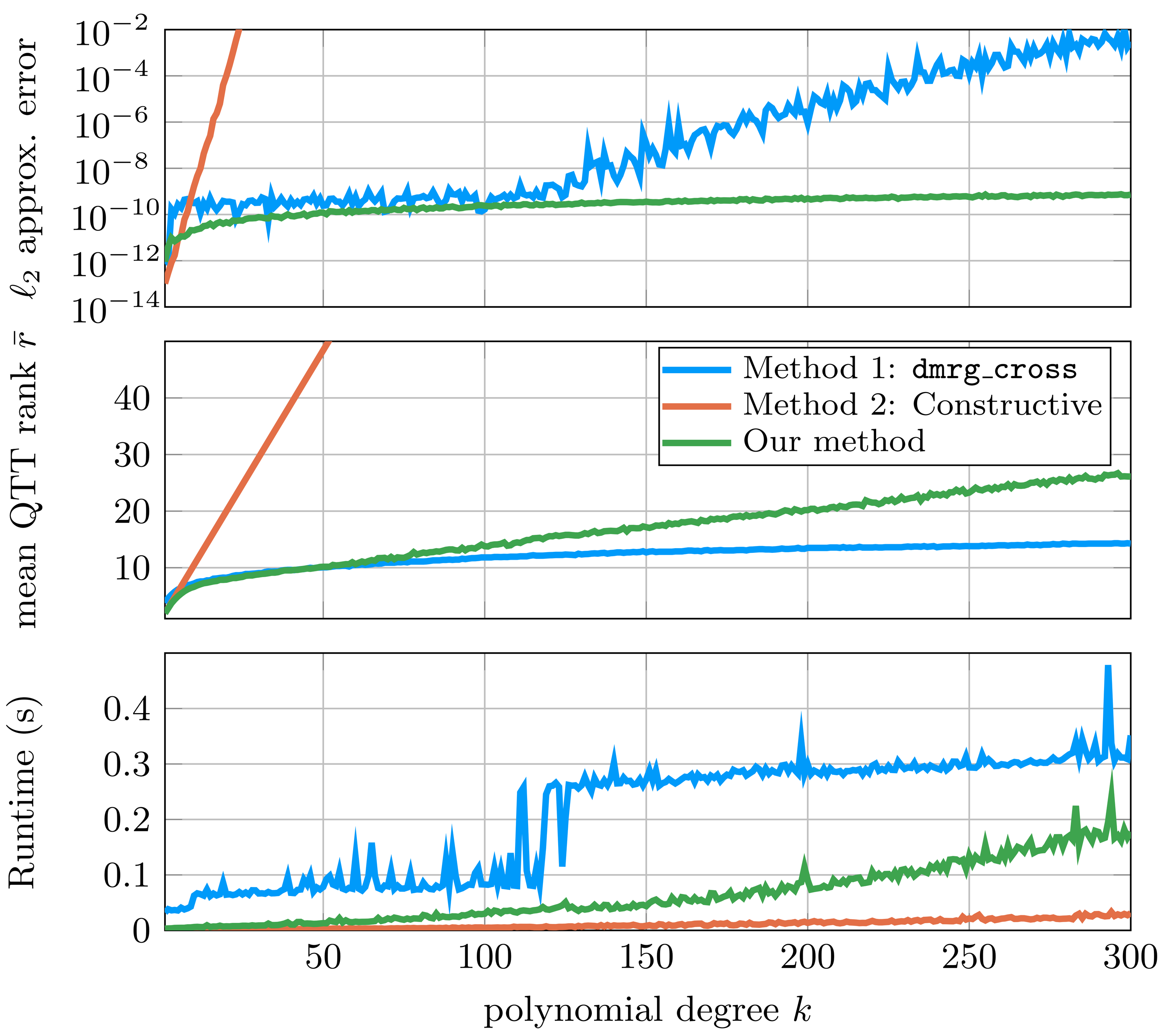
$$p(x^{(1)}_{i_1}+x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) = \mathbf{P}(i_1,\cdots,i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)\cdots\mathbf{G}_d(i_d) \text{ for } i_1,\cdots,i_d = 1,2.$$

- Storage of U_ℓ in QTT: $\mathcal{O}(DRdr^2) = \mathcal{O}(DRr^2\log n)$.
- **Classical result**: QTT ranks of degree m polynomials $\le m+1$ [Khoromskij'11, Oseledets'13] (numerically even $\log m$) $\rightsquigarrow$ reduces storage of U_ℓ from $\mathcal{O}(DRn)$ 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 $\rightsquigarrow$ our focus

Numerical experiments

Approximating Chebyshev polynomials $T_k(x)$ in QTT

- Similar results are obtained for other class of polynomials, e.g., random linear combinations of monomials/Chebyshev polynomials.



Approximation of a polynomial in QTT format

Computational methods for the TT format of $\mathbf{P}$

1. **Direct method**: apply directly the adaptive TT cross, e.g., `dmrg_cross` [Savostyanov/Oseledets'11].
   - Disadvantage: heuristic; doesn't exploit the 1D nature of the problem, just view $\mathbf{P}$ as a black box tensor.
2. **Constructive method**: $\exists$ an analytic formula [Oseledets'13] for the cores $\mathbf{G}_\ell(i_\ell)\in\mathbb{R}^{(m+1)\times(m+1)}$.
   - Disadvantage: is numerically unstable even for moderate $m>20$ & produces pessimistic TT ranks $m+1$.

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

Recall Oseledets' constructive method for $p(x) = p_0+p_1x+\cdots+p_mx^m$

Introduce the notation: $X_{\le k}(x) := [1,x,\cdots,x^k]^\top\in\mathbb{R}^{k+1}$

1. Notice that with $M(\alpha+1,\beta+1) = p_{\alpha+\beta}C^{\alpha}_{\alpha+\beta}$ for $\alpha+\beta\le m$ otherwise 0 (M is skew-upper triangular), it holds

$$p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M(\alpha+1,\beta+1)x^\alpha y^\beta \implies p(x+y) = X_{\le m}(x)^\top M X_{\le m}(y). \quad (3)$$

2. Let $x = x_{i_1}$ and $y = x_{i_2}+\cdots+x_{i_d}$, then separate x_{i_1} from the rest:

$$p(x^{(1)}_{i_1}+x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) = \underbrace{X_{\le m}(x^{(1)}_{i_1})^\top M}_{=\mathbf{G}_1(i_1)} X_{\le m}(x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}). \quad (4)$$

3. Similarly, $\mathbf{G}_2(i_2)$ is constructed such that $X_{\le m}(x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) = \mathbf{G}_2(i_2)X_{\le m}(x^{(3)}_{i_3}+\cdots+x^{(d)}_{i_d})$ holds,

$$\implies \mathbf{P}(i_1,i_2,\cdots,i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2)X_{\le m}(x^{(3)}_{i_3}+\cdots+x^{(d)}_{i_d}).$$

4. Other cores $\mathbf{G}_3(i_3),\cdots,\mathbf{G}_d(i_d)$ are constructed similarly.

Disadvantage: Unstable since M contains large binomial coefficients & small powers e.g. $(x^{(1)}_{i_1})^\alpha$
Fixing idea: Replace x^α, y^β by Chebyshev polynomials!

$$p(x+y) = \sum_{\alpha=0}^{m}\sum_{\beta=0}^{m} M^c(\alpha+1,\beta+1)T_\alpha(x)T_\beta(y) \quad (5)$$

- A subtle but crucial difference between (3) and (5): (3) holds for all $x,y\in\mathbb{R}$ for fixed coefficients M, while (5) only holds for $x\in I_x$ and $y\in I_y$ with some intervals I_x, I_y and the coefficients M^c depend accordingly on these intervals.
- For an interval $I=[a,b]$, introduce the notation:
$T^I_\ell(x) := T_\ell((2x-(a+b))/(b-a))$, $T^I_{\le k}(x) := [1, T^I_1(x),\cdots,T^I_k(x)]^\top\in\mathbb{R}^{k+1}$ for $x\in I=[a,b]$.

Our novel method: constructive, stable & rank adaptive

Recall: 1D Chebyshev interpolation

Let $p: I\to\mathbb{R}$ be a univariate function. Then its Chebyshev interpolation of degree m is given by

$$p(x)\approx g_m := \sum_{\ell=0}^{m} c_\ell T^I_\ell(x) = \mathbf{c}^\top T^I_{\le m}(x) \text{ with } \mathbf{c} = W\mathbf{p},\ W \text{ being an inverse DCT matrix}$$

and $\mathbf{p}\in\mathbb{R}^{m+1}$ containing the function values of p at the Chebyshev points on I. In particular if p is a polynomial of degree $\le m$, then the interpolation is **exact**: $p(x) = g_m(x)$ for all $x\in I$.

- Let $I_{\ge k} = [\sum_{\ell=k}^{d} x^{(\ell)}_1, \sum_{\ell=k}^{d} x^{(\ell)}_2]$, then we have $x^{(k)}_{i_k}+x^{(k+1)}_{i_{k+1}}+\cdots+x^{(d)}_{i_d}\in I_{\ge k}$.

1. Let $M^c_{i_1}\in\mathbb{R}^{1\times(m+1)}$ be the Chebyshev coefficients of the function $I_{\ge 2}\ni x\mapsto p(x^{(1)}_{i_1}+x)$:

$$p(x^{(1)}_{i_1}+x) = M^c_{i_1}T^{I_{\ge 2}}_{\le m}(x) \text{ for } x\in I_{\ge 2},\ i_1=1,2, \text{ and in particular} \quad (6)$$
$$\mathbf{P}(i_1,i_2,\cdots,i_d) = p(x^{(1)}_{i_1}+x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) = M^c_{i_1}T^{I_{\ge 2}}_{\le m}(x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) \quad (7)$$

We can already set here $\mathbf{G}(i_1) := M^c_{i_1}$ to obtain the first QTT core (c.f. Eq. (4))

2. Yet to make it rank adaptive, we compute a LRA (e.g. SVD) of the Chebyshev coefficients:

$$\begin{bmatrix} M^c_1 \\ M^c_2\end{bmatrix} \approx U_1V_1^\top \text{ with } U_1\in\mathbb{R}^{2\times r_1},\ V_1\in\mathbb{R}^{(m+1)\times r_1}.$$

3. Let $U_{1,i_1} := U_1(i_1,:)\in\mathbb{R}^{1\times r_1}$, then $M^c_{i_1}\approx U_{1,i_1}V_1^\top$. Then Eq. (7) becomes

$$\mathbf{P}(i_1,i_2,\cdots,i_d)\approx U_{1,i_1}V_1^\top T^{I_{\ge 2}}_{\le m}(x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) = U_{1,i_1}v^{(2)}_{\le r_1}(x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) \quad (8)$$

where we have defined the polynomial vector $v^{(2)}_j(y)$ on the interval $I_{\ge 2}$ as

$$v^{(2)}_{\le r_1}(y) := [v^{(2)}_1(y),\cdots,v^{(2)}_{r_1}(y)]^\top\in\mathbb{R}^{r_1} \text{ with } v^{(2)}_j(y) := V_1(:,j)^\top T^{I_{\ge 2}}_{\le m}(y) \text{ for } j=1,\cdots,r_1,\ y\in I_{\ge 2}.$$

The first QTT core can thus be taken as $\mathbf{G}_1(i_1) = U_{1,i_1}$.

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

$$v^{(2)}_{\le r_1}(x^{(2)}_{i_2}+\cdots+x^{(d)}_{i_d}) = C^c_{2,i_2}T^{I_{\ge 3}}_{\le m}(x^{(3)}_{i_3}+\cdots+x^{(d)}_{i_d}). \quad (9)$$

We then compute a LRA of $\begin{bmatrix} C^c_{2,1}\\ C^c_{2,2}\end{bmatrix}\approx U_2V_2^\top$ with $U_2\in\mathbb{R}^{2r_1\times r_2}$, $V_2\in\mathbb{R}^{(m+1)\times r_2}$.

Let us denote $U_{2,i_2} := U_2(r_1(i_2-1)+1:r_1i_2,:)\in\mathbb{R}^{r_1\times r_2}$, then $C^c_{2,i_2}\approx U_{2,i_2}V_2^\top$. Thus

$$\mathbf{P}(i_1,i_2,\cdots,i_d)\approx U_{1,i_1}C^c_{2,i_2}T^{I_{\ge 3}}_{\le m}(x^{(3)}_{i_3}+\cdots+x^{(d)}_{i_d})\approx U_{1,i_1}U_{2,i_2}V_2^\top T^{I_{\ge 3}}_{\le m}(x^{(3)}_{i_3}+\cdots+x^{(d)}_{i_d}) \quad (10)$$

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

References

[1] 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.
[2] P. Benner, B. Khoromskij, B. Sun. *Bridging functional and discrete tensor representation of multivariate functions using QTT*, in preparation, 2025.
[3] H. Behnam, L. N. Trefethen. *Chebfun in three dimensions*, SIAM J. Sci. Comput., 2017.