Approximation to multivariate functions in the extended functional tensor train format

Christoph Strössner, Bonan Sun and Daniel Kressner

École Polytechnique Fédérale de Lausanne (EPFL), Institute of Mathematics, Switzerland. Contact: bonan.sun@epfl.ch

Functional tensor train (FTT) approximation

A continuous version of TT format for $f : [-1,1]^d \to \mathbb{R}$, i.e., FTT format:

$$f(x_1,\dots,x_d) \approx \sum_{\alpha_1=1}^{R_1} \cdots \sum_{\alpha_{d-1}=1}^{R_{d-1}} g^{(1)}_{1,\alpha_1}(x_1) g^{(2)}_{\alpha_1,\alpha_2}(x_2) \cdots g^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(x_{d-1}) g^{(d)}_{\alpha_{d-1},1}(x_d). \quad (1)$$

Goal: Approximate f in FTT format (1) using as few function evaluations as possible.

FTT via tensorized Chebyshev interpolation + TT

Define the function evaluation tensor $\mathcal{T} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ on the tensorized Chebyshev nodes:

$$\mathcal{T}_{i_1,\dots,i_d} = f(x^{(1)}_{i_1},\dots,x^{(d)}_{i_d}),\ x^{(\ell)}_k = \cos(\pi k/(n_\ell - 1)),\ k = 1,\dots,n_\ell,\ \ell = 1,\dots,d.$$

Then the Chebyshev interpolant $\tilde{f}$ of f is given by

$$f(x_1,\dots,x_d) \approx \tilde{f}(x_1,\dots,x_d) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathcal{A}_{i_1,\dots,i_d} T_{i_1}(x_1)\cdots T_{i_d}(x_d),$$

where $T_k(x)$ is the k-th Chebyshev polynomial and the coefficient tensor $\mathcal{A}$ is given by

$$\mathcal{A} = \mathcal{T} \times_1 F^{(1)} \times_2 F^{(2)} \times_3 \cdots \times_d F^{(d)} \text{ for some DCT matrices } F^{(\ell)} \in \mathbb{R}^{n_\ell \times n_\ell}.$$

Observation: $\mathcal{T} \approx \hat{\mathcal{T}}$ in TT format leads to $\tilde{f} \approx \hat{f}$ in FTT format:

$$\hat{\mathcal{T}}_{i_1,\dots,i_d} = \sum_{\alpha_1=1}^{R_1} \cdots \sum_{\alpha_{d-1}=1}^{R_{d-1}} \mathcal{G}^{(1)}_{1,i_1,\alpha_1} \mathcal{G}^{(2)}_{\alpha_1,i_1,\alpha_2} \cdots \mathcal{G}^{(d-1)}_{\alpha_{d-2},i_{d-1},\alpha_{d-1}} \mathcal{G}^{(d)}_{\alpha_{d-1},i_d,1}, \quad (2)$$

$$\Longrightarrow \hat{f} \text{ in FTT format (1) with } g^{(\ell)}_{\alpha_{\ell-1},\alpha_\ell}(x) = \sum_{j=1}^{n_\ell} \sum_{k=1}^{n_\ell} F^{(\ell)}_{j,k} \mathcal{G}^{(\ell)}_{\alpha_{\ell-1},k,\alpha_\ell} T_j(x).$$

Updated Goal: Approximate $\mathcal{T}$ in TT format (2) by accessing as few entries as possible.

Extended (functional) tensor train format (EFTT)

Novel idea: Approximate $\mathcal{T}$ in Tucker format first $\mathcal{T} \approx \mathcal{C} \times_1 U^{(1)} \times_2 \cdots \times_d U^{(d)}$ and then approximate the Tucker core $\mathcal{C} \in \mathbb{R}^{r_1\times\cdots\times r_d}$ in TT format with TT cores $\mathcal{H}^{(\ell)} \in \mathbb{R}^{R_{\ell-1}\times r_\ell \times R_\ell}$. In this case, $\mathcal{T}$ is approximated in the extended TT format:

$$\mathcal{T}_{i_1,\dots,i_d} \approx \hat{\mathcal{T}}_{i_1,\dots,i_d} = \sum_{j_1=1}^{r_1} \cdots \sum_{j_d=1}^{r_d} \sum_{\alpha_1=1}^{R_1} \cdots \sum_{\alpha_{d-1}}^{R_{d-1}} \mathcal{H}^{(1)}_{1,j_1,\alpha_1} \cdots \mathcal{H}^{(d)}_{\alpha_{d-1},j_d,1} U^{(1)}_{i_1,j_1} \cdots U^{(d)}_{i_d,j_d}. \quad (3)$$

Reduce space complexity $\mathcal{O}(dnR^2)$ of TT to $\mathcal{O}(drR^2 + dnr)$ especially when $r \ll n$.

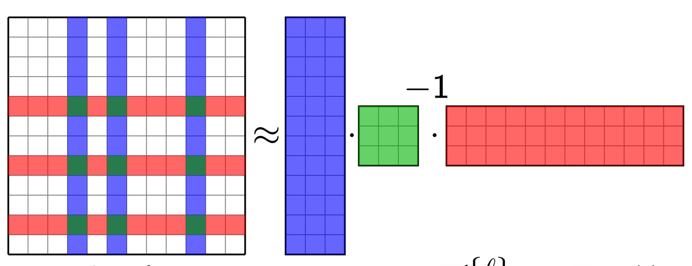
Updated Goal: Approximate $\mathcal{T}$ in ETT format (3) by accessing as few entries as possible.

Novel EFTT approximation algorithm — Phase 1

Phase 1: Construct $U^{(\ell)} \in \mathbb{R}^{n_\ell \times r_\ell}$ by accessing as few entries of $\mathcal{T}$ as possible.

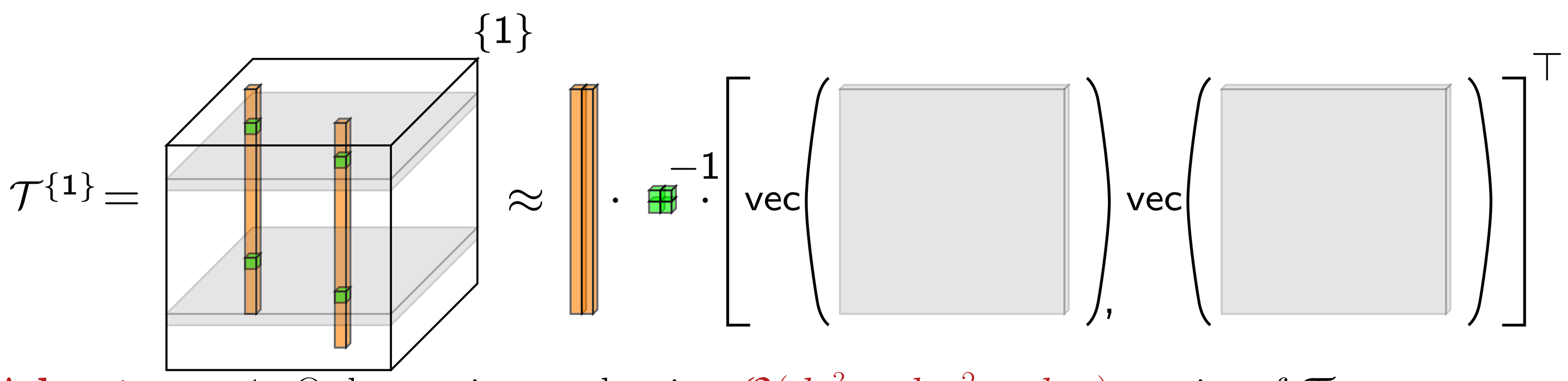
- $U^{(\ell)}$ should approximate the space spanned by the mode-ℓ fibers of $\mathcal{T}$.
- We propose the randomized pivoted adaptive cross approximation (RPACA) algorithm:

$$M \approx M(:,J)M(I,J)^{-1}M(I,:) \text{ for matrix } M \text{ and index sets } I, J.$$



- Applying RPACA to the mode-ℓ matricization $\mathcal{T}^{\{\ell\}} \in \mathbb{R}^{n_\ell \times n_1\cdots n_{\ell-1}n_{\ell+1}\cdots n_d}$ of $\mathcal{T}$ yields

$$\mathcal{T}^{\{\ell\}} \approx \mathcal{T}^{\{\ell\}}(:,J_\ell)(\mathcal{T}^{\{\ell\}}(I_\ell,J_\ell))^{-1}\mathcal{T}^{\{\ell\}}(I_\ell,:), \text{ let } U^{(\ell)} \text{ be } \mathcal{T}^{\{\ell\}}(:,J_\ell).$$



Advantages:
1. Only requires evaluating $\mathcal{O}(dr^3 + dsr^2 + dnr)$ entries of $\mathcal{T}$.
2. Adaptivity in Tucker rank r_ℓ's naturally (by choosing ACA tol ε).
3. Adaptivity in polynomial degree n_ℓ's by leveraging **Chebfun** heuristics [1].

References

[1] J. L. Aurentz and L. N. Trefethen, *Chopping a Chebyshev series*, ACM Trans. Math. Software, 43 (2017), pp. 1–21.
[2] D. V. Savostyanov, *Quasioptimality of maximum-volume cross interpolation of tensors*, Linear Algebra Appl., 458 (2014), pp. 217–244.
[3] A. Gorodetsky, *Continuous low-rank tensor decompositions*, PhD thesis, MIT, Cambridge, MA, 2017.

Novel EFTT approximation algorithm — Phase 2 & 3

Phase 2: Construct $\mathcal{C}$ implicitly, i.e., define a procedure to compute any entry of $\mathcal{C}$.

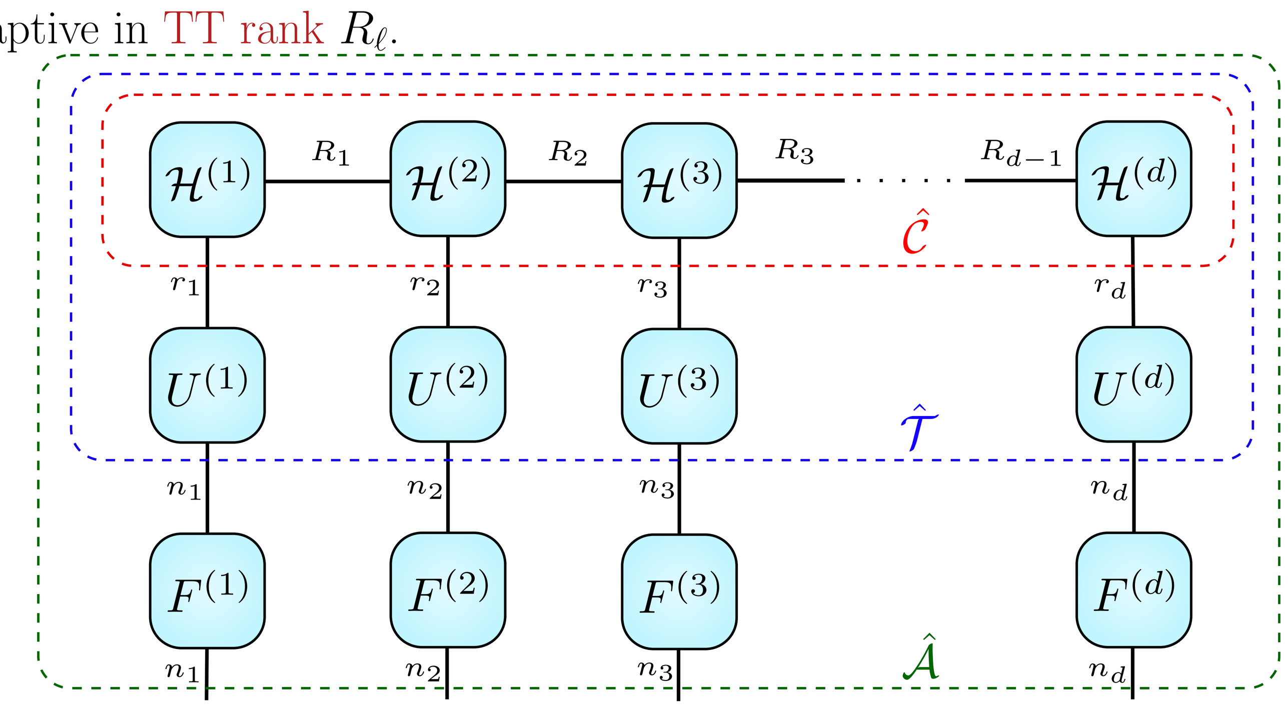
- Best possible $\mathcal{C}$ given $U^{(\ell)}$ is by orthogonally projecting $\mathcal{T}$ onto the span of $U^{(\ell)}$'s, i.e. multiplying $\mathcal{T}$ with $Q^{(\ell)}(Q^{(\ell)})^\top$ in each mode, where $Q^{(\ell)}$ is the thin QR of $U^{(\ell)}$.
- Main issue: requires full evaluation of $\mathcal{T}$, which is not feasible.
- Our solution: use oblique projection $Q^{(\ell)}(\Phi_{I_\ell}^\top Q^{(\ell)})^{-1}\Phi_{I_\ell}^\top$ for some sampling matrix Φ_{I_ℓ}, i.e., $\Phi_{I_\ell}^\top Q^{(\ell)} = Q^{(\ell)}(I_\ell,:)$.

$$\mathcal{T} \approx \underbrace{(\mathcal{T} \times_1 \Phi_{I_1}^\top \times_2 \cdots \times_d \Phi_{I_d}^\top)}_{\mathcal{C}=\mathcal{T}(I_1,\cdots,I_d)} \times_1 \underbrace{Q^{(1)}(\Phi_{I_1}^\top Q^{(1)})^{-1}}_{\text{updated } U^{(1)}} \times_2 \cdots \times_d \underbrace{Q^{(d)}(\Phi_{I_d}^\top Q^{(d)})^{-1}}_{\text{updated } U^{(d)}}$$

- Error introduced by the oblique projection depends on $\|Q^{(\ell)}(I_\ell,:)^{-1}\|_2$, which can be minimized by selecting I_ℓ carefully, e.g., using model order reduction methods like **DEIM**.
- No evaluation of $\mathcal{T}$ in Phase 2, just update $U^{(\ell)}$ and define $\mathcal{C}$ as a subtensor of $\mathcal{T}$ implicitly.

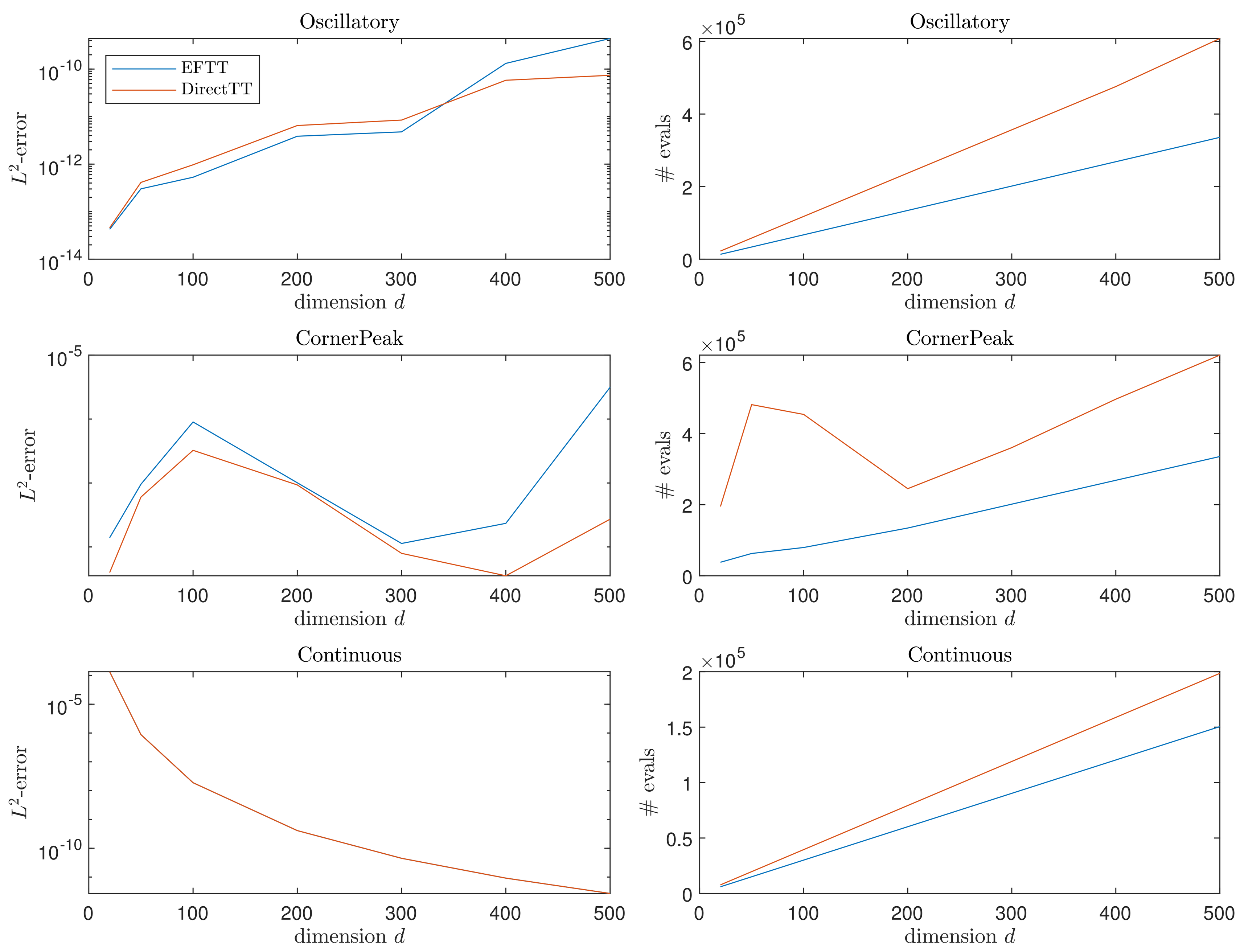
Phase 3: Construct the TT format of $\mathcal{C}$ by accessing as few entries of $\mathcal{C}$ as possible.

- We use the **greedy2cross** algorithm [2], requires $\mathcal{O}(drR^2)$ function evaluations.
- It is adaptive in TT rank R_ℓ.



Numerical results

Comparison of EFTT with directly applying **greedy2cross** to $\mathcal{T}$ for Genz functions:



Comparison of EFTT with continuous analog of TT-cross proposed in [3] (c3py) for functions tested in [3] with various application backgrounds:

Function (dimension)	Algorithm	Error	# function evals	# dofs (storage)	$\max_\ell n_\ell$	$\max_\ell R_\ell$	$\max_\ell r_\ell$
Piston (7)	EFTT	3.71e-09	174188	69019	33	23	11
	c3py	3.85e-05	251760	66080	35	24	
Borehole (8)	EFTT	3.95e-02	6552	1116	32	2	4
	c3py	2.08e-03	14346	577	70	2	
OTL Circuit (6)	EFTT	7.93e-11	6670	1083	27	5	5
	c3py	4.07e-08	15674	1782	28	5	
Robot Arm (8)	EFTT	8.12e-02	499954	54760	94	12	27
	c3py	3.85e-01	2018017	228439	105	20	
Wing Weight (10)	EFTT	2.83e-14	2867	560	24	2	2
	c3py	2.15e-13	12224	554	19	2	
Friedman (5)	EFTT	2.16e-11	5238	404	19	3	4
	c3py	8.08e-05	12142	710	15	4	
G & L (6)	EFTT	4.95e-06	1547	356	29	2	2
	c3py	3.51e-02	13928	374	105	2	
G & P 8D (8)	EFTT	4.77e-11	19527	3902	24	6	7
	c3py	9.54e-10	27336	5136	21	7	
D & P Exp (3)	EFTT	1.13e-14	2404	646	105	2	2
	c3py	4.78e-10	12162	336	49	2	