Outline of Lecture 2 (I).

1. Wide range applications in $\mathbb{R}^d$.

2. $d = 2$: Main properties of the rank-R matrices.

3. Approximation by low rank matrices: Truncated SVD, reduced SVD, and adaptive cross approximation (ACA).

4. $\mathcal{H}$-matrices in dimension ≤ 3: advantages and limitations.

5. FFT, FFT$_d$, and circulant convolution.

6. A paradigm of super-computing:

increase in computer power does not relax the curse of dimensionality.

- Elliptic (parameter-dependent) eq.: Find $u \in H_0^1(\Omega)$, s.t.,

$$\mathcal{H}u := -\operatorname{div}(A \operatorname{grad} u) + Vu = F \quad \text{in} \quad \Omega \in \mathbb{R}^d.$$

- EVP: Find a pair $(\lambda, u) \in \mathbb{R} \times H_0^1(\Omega)$, s.t., $\langle u, u \rangle = 1$, and

$$\mathcal{H}u = \lambda u \quad \text{in } \Omega \in \mathbb{R}^d,$$
$$u = 0 \qquad \text{on } \partial\Omega.$$

- Parabolic equations: Find $u : \mathbb{R}^d \times (0, \infty) \to \mathbb{R}$, s.t.

$$u(x,0) \in H^2(\mathbb{R}^d): \quad \sigma \frac{\partial u}{\partial t} + \mathcal{H}u = 0, \quad \mathcal{H} = \Delta_d + V(x_1, ..., x_d).$$

Specific features:

▷ High spacial dimension: $\Omega = (-b, b)^d \in \mathbb{R}^d$ ($d = 2, 3, ..., 100, ...$).

▷ Multiparametric eq.: $A(y, x)$, $u(y, x)$, $y \in \mathbb{R}^M$ ($M = 1, 2, ..., 100, ..., \infty$).

▷ Nonlinear, nonlocal (integral) operator $V = V(x, u)$, singular potentials.

General mathematical tasks

- Fast Poisson solver, preconditioning $\Rightarrow (-\Delta + I)^{-1}$.
- Convolution transform in $\mathbb{R}^d$ with Green's function for d-Laplacian ($d \geq 3$),

$$f(x) = \int_{\mathbb{R}^d} \frac{\rho(y)}{\|x-y\|^{d-2}} dy, \quad x \in \mathbb{R}^d.$$

  $O(dn \log n)$-algorithms, numerics in electronic structure calculations.
- Parabolic eqs (heat transfer, molecular dynamics, ...)
  $\frac{\partial u}{\partial x} + Au = f \quad \Rightarrow \quad \exp(-tA)$, Cayley Transform $\frac{I+A}{I-A}$.
- Multilinear algebra (MLA), complexity theory (e.g., Strassen's algorithm by tensor decomposition).
- Matrix product states (TT, TC, QTT) + DMRG-type iteration for slightly entangled sytems (electronic structure, molecular dynamics, quantum computing).

Many-particle models

- Hartree-Fock equation

$$\left[-\frac{1}{2}\Delta - V_c(x) + \int_{\mathbb{R}^3} \frac{\rho(y,y)}{\|x-y\|} dy\right] \phi(x) - \frac{1}{2}\int_{\mathbb{R}^3} \frac{\rho(x,y)}{\|x-y\|} \phi(y) dy = \lambda \phi(y),$$

  $\rho(x,y) = \sum_{i=1}^{N_e/2} \phi_i(x)\phi_i(y)$ electron density matrix,
  $e^{-\mu\|x\|}$ - density function for hydrogen atom,
  $\frac{1}{\|x\|}$ - Newton potential,
  V_c - external potential with singularities at centers of atoms.
  Tensor approximation scheme and numerics Lect. 4.
- Kohn-Sham equation (simplyfied Hartree-Fock eq.)

$$\left[-\frac{1}{2}\Delta - V_c(x) + \int_{\mathbb{R}^3} \frac{\rho(y)}{\|x-y\|} dy - \alpha V_\rho(x)\right] \psi = \lambda \psi, \quad V_\rho(x) = \left\{\frac{3}{\pi}\rho(x)\right\}^{1/3}.$$

- Poisson-Boltzmann eq. (the electrostatic potential of proteins)

$$\nabla \cdot [\varepsilon(x) \nabla \cdot \phi(x)] - \varepsilon(x) h(x)^2 sinh[\phi(x)] + 4\pi\rho(x)/kT = 0, \quad x \in \mathbb{R}^3.$$

  If $\varepsilon(x) = \varepsilon_0$, $h(x) = h$, $\rho(x) = \delta(x)$, then $\phi(x) = \frac{e^{-h\|x\|}}{\|x\|}$.

Parametric Elliptic Problems: Stochastic PDEs

Find $u_M \in L^2(\Gamma) \times H_0^1(D)$, s.t.

$$\mathcal{A} u_M(y,x) = f(x) \quad \text{in } D, \quad \forall y \in \Gamma,$$
$$u_M(y,x) = 0 \quad \text{on } \partial D, \quad \forall y \in \Gamma,$$

$$\mathcal{A} := -\operatorname{div}\left(a_M(y,x)\operatorname{grad}\right), \quad f \in L^2(D), \quad D \in \mathbb{R}^d, \quad d = 1,2,3,$$

$a_M(y,x)$ is smooth in $x \in D$, $y = (y_1, ..., y_M) \in \Gamma := [-1,1]^M$, $M \leq \infty$.

Additive case (via the truncated Karhunen-Loéve expansion)

$$a_M(y,x) := a_0(x) + \sum_{m=1}^{M} a_m(x) y_m, \quad a_m \in L^\infty(D), \quad M \to \infty.$$

Log-additive case

$$a_M(y,x) := \exp(a_0(x) + \sum_{m=1}^{M} a_m(x) y_m) > 0.$$

- Computing the truncated Karhunen-Loéve expansion.
- Analysis of best N-term approximations.
- Tensor representation of stochastic-Galerkin and collocation matrices.
- Tensor truncated preconditioned iteration.

Matrix SVD

Lem. 2.1. (matrix SVD). Every real (complex) $\tau \times \sigma$-matrix M can be representd as the product

$$M = U \cdot \mathbf{S} \cdot V^T := \mathbf{S} \times_1 U \times_2 V \equiv \mathbf{S} \times_1 U^{(1)} \times_2 U^{(2)},$$

in which

1. $U^{(1)} = [U_1^{(1)} U_2^{(1)} ... U_\tau^{(1)}]$ is a unitary $\tau \times \tau$-matrix,
2. $U^{(2)} = [U_1^{(2)} U_2^{(2)} ... U_\sigma^{(2)}]$ is a unitary $\sigma \times \sigma$-matrix,
3. $\mathbf{S}$ is an $\tau \times \sigma$-matrix (core tensor) with the properties of

$$(i)\ pseudodiagonality: \quad \mathbf{S} = diag\{\sigma_1, \sigma_2, ..., \sigma_{\min(\tau,\sigma)}\},$$

$$(ii)\ ordering: \quad \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{\min(\tau,\sigma)} \geq 0.$$

The σ_i are singular values of M, and the vectors $U_i^{(1)}$ and $U_i^{(2)}$ are, resp., an ith left and ith right singular vectors.

The class of rank $\leq k$ matrices in $\mathbb{R}^{\tau\times\sigma}$ will be called by $\mathcal{R}_k$-matrices, i.e. $rank(M) \leq k$ for $M \in \mathcal{R}_k$.

Each $M \in \mathcal{R}_k$ can be represented in the form

$$M = A \cdot B^T, \qquad A \in \mathbb{R}^{\tau\times k}, \quad B \in \mathbb{R}^{\sigma\times k}. \tag{1}$$

Lem. 2.2. Attractive features of $\mathcal{R}_k$-matrices:

1. The set $\mathcal{R}_k$ is closed (nontrivial result in linear algebra).

2. Only $k(\tau+\sigma)$ numbers are required to store an $\mathcal{R}_k$-matrix.

3. The matrix-vector multiplication $x \mapsto y := Mx$, $x \in \mathbb{R}^\sigma$ can be done in two steps:

$y' := B^T x \in \mathbb{R}^k$, and $y := Ay' \in \mathbb{R}^\tau$.

The corresponding cost is $2k(\sigma+\tau)$.

4. The sum of two $\mathcal{R}_k$-matrices $R_1 = A_1 B_1^T$, $R_2 = A_2 B_2^T$ is an $\mathcal{R}_{2k}$-matrix,

$$R_1 + R_2 = [A_1|A_2][B_1|B_2]^T, \quad [A_1|A_2] \in \mathbb{R}^{\tau\times 2k}, \quad [B_1|B_2] \in \mathbb{R}^{\sigma\times 2k}.$$

5. The multiplication of $R \in \mathcal{R}_k$ by an arbitrary matrix M of the proper size gives again an $\mathcal{R}_k$-matrix:

$$RM = A(M^T B)^T, \qquad MR = (MA)B^T.$$

6. The best approximation of an arbitrary matrix $M \in \mathbb{R}^{\tau\times\sigma}$ by an $\mathcal{R}_k$-matrix M_k, say in the Frobenius norm, that is

$$\|A\|_F^2 := \sum_{(i,j)\in\tau\times\sigma} a_{ij}^2,$$

can be calculated by the truncated SVD (discrete version of the Schmidt decomposition).

Alg. 2.1. (Truncated SVD). For given $k \in \mathbb{N}$, let $M = U\Sigma V^T$ be the SVD of M, i.e., $\Sigma = diag\{\sigma_1, \ldots \sigma_k, ..., \sigma_n\}$, $n = \min(\tau, \sigma)$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$, and $U = [U_1, ..., U_k, U_{k+1}, ..., U_\tau]$, $V = [V_1, ..., V_k, V_{k+1}, ..., V_\sigma]$. Set $\Sigma_k := diag\{\sigma_1, \ldots, \sigma_k, 0, \ldots, 0\}$,

$$M_k := U\Sigma_k V^T \equiv \bar{U}\bar{\Sigma}_k \bar{V}^T \approx M,$$

$$\|M_k - M\|_F \leq \sqrt{\sum_{j=k+1}^{n} \sigma_j^2}.$$

The complexity of the truncated SVD: $\mathcal{O}(\tau\sigma^2)$ with $\tau \geq \sigma$.

Too expensive for large τ and σ.

Is it possible to compute almost the best rank-k matrix approximation getting rid of full matrix SVD ? – Yes.

If $M \in \mathcal{R}_m$, then its best approximation $M_k \in \mathcal{R}_k$, $k < m$, can be computed by the following QR-SVD scheme.

Alg. 2.2. (Reduced truncated SVD). Given $M = AB^T \in \mathcal{R}_m$,

(i) Calculate the QR-decompositions $A = Q_A R_A$ and $B = Q_B R_B$, with the unitary matrices $Q_A \in \mathbb{R}^{\tau \times m}$, and $Q_B \in \mathbb{R}^{\sigma \times m}$, and upper triangular matrices $R_A, R_B \in \mathbb{R}^{m \times m}$.

(ii) Calculate a SVD, $R_A R_B^T = U\Sigma V^T$ (with the cost $O(m^3)$).

(iii) Define $M_k = A_k B_k^T$ with $A_k := Q_A U_k \Sigma_k \in \mathbb{R}^{\tau \times k}$ and $B_k := Q_B V_k \in \mathbb{R}^{\sigma \times k}$, where $U_k := [U_1, \ldots, U_k]$, $V_k := [V_1, \ldots, V_k]$ (in both cases, first k columns) and the truncated matrix Σ_k of Σ are defined by truncated SVD of $R_A R_B^T = U\Sigma V^T$.

Alg. 2.2 can be implemented in $\mathcal{O}(m^2(\tau + \sigma) + m^3)$ operations.

Exer. 2.1 Compute the rank-r, $r = 2M + 1$, sinc quadrature (**[7], L. 1**) approximation of the Hilbert matrix $A = \{a_{ij}\}$, $(i, j = 1, ..., n)$

$$a_{ij} = 1/(i + j) = \int_0^\infty e^{-(i+j)t}\, dt \approx \sum c_k e^{-(i+j)t_k},$$

for $n = 10^3$, 10^4, and $M = 64$. Apply to the result the best low rank approximation via reduced truncated SVD by Alg. 2.2.

Adaptive cross approximation (ACA)

In FEM/BEM applications, nearly best (suboptimal) rank-k approximation over partial data can be computed by the heuristic method called adaptive cross approximation (ACA),

cf. [3], [6], E. Tyrtyshnikov et al.

Many matrix decomposition algorithms can be represented as a sequence of rank-one *Wedderburn updates.*

J. H. M. Wedderburn, Lectures on matrices, colloquim publications, vol. XVII, AMS, NY, 1934.

For a given $m \times n$ matrix A and vectors x, y of appropriate sizes, s.t. $x^T Ay \neq 0$, matrix

$$B = A - \frac{Ayx^T A}{x^T Ay},$$

has rank(B) =rank$(A) - 1$. For the rank-r matrix $A_0 = A$ after r updates (if do not fail) of form

$$A_k = A_{k-1} - \frac{A_{k-1} y_k x_k^T A_{k-1}}{x_k^T A_{k-1} y_k}, \quad \text{with} \quad x_k^T A_{k-1} y_k \neq 0,$$

the matrix A_r becomes zero leading to rank-r decomposition of A.

Adaptive cross approximation (ACA)

Sketch of the ACA:

- Starting from $R_0 = A \in \mathbb{R}^{m\times n}$, find a nonzero pivot in R_k, say (i_k, j_k), and subtract a scaled outer product of the i_kth row and the j_kth column:

$$R_{k+1} := R_k - \frac{1}{(R_k)_{i_k j_k}} u_k v_k^T, \quad \text{with} \quad u_k = (R_k)_{1:m, j_k},\ v_k = (R_k)_{i_k, 1:n},$$

where we use the notation $(R_k)_{i_k,1:n}$ and $(R_k)_{1:m,j_k}$ for the i_kth row and the j_kth column of R_k, respectively.

- j_k is chosen as the maximum element in modulus of the i_kth row, i.e.,

$$|(R_{k-1})_{i_k j_k}| = \max_{j=1,\ldots,n} |(R_{k-1})_{i_k j}|.$$

The choice of i_k will be similar.

- The matrix $S_r := \sum_{k=1}^r u_k v_k^T$ will be used as the rank-r approximation of $A = S_r + R_r$, since $rank(S_r) \leq r$.

- Apply the reduced truncated SVD to S_r for the rank optimization.

Rem. 2.1. SPD case: ACA = Pivoted Cholesky decompositions !

$\mathcal{H}$-matrix format: brief survey

$\mathcal{H}$- and $\mathcal{H}^2$-matrix technique is a direct descendant of *panel clustering, fast multipole and mosaic-skeleton approximations*.

In addition, it allows data-sparse matrix-matrix operations.

$\mathcal{M}_{\mathcal{H},k}(T_{I\times I}, \mathcal{P})$, *the class of data-sparse hierarchical $\mathcal{H}$-matrices* - **Hackbusch, Khoromskij, Bebendorf, Börm, Grasedyck, Sauter ('99 - '05)**.

The construction of $\mathcal{H}$-matrices defined on the product index set $I \times I$, is based on the following ingredients:

- An $\mathcal{H}$-tree $T(I)$ of the index set I (hierarchical cluster tree).
- The admissible partitioning $\mathcal{P}$ of $I \times I$ based on a block cluster tree $T(I \times I)$.
- Low rank approximation of all large enough blocks in $\mathcal{P}$.

Examples of hierarchical partitioning

Hierarchical Partitionings $\mathcal{P}_{1/2}(I \times I)$ and $\mathcal{P}_W(I \times I)$

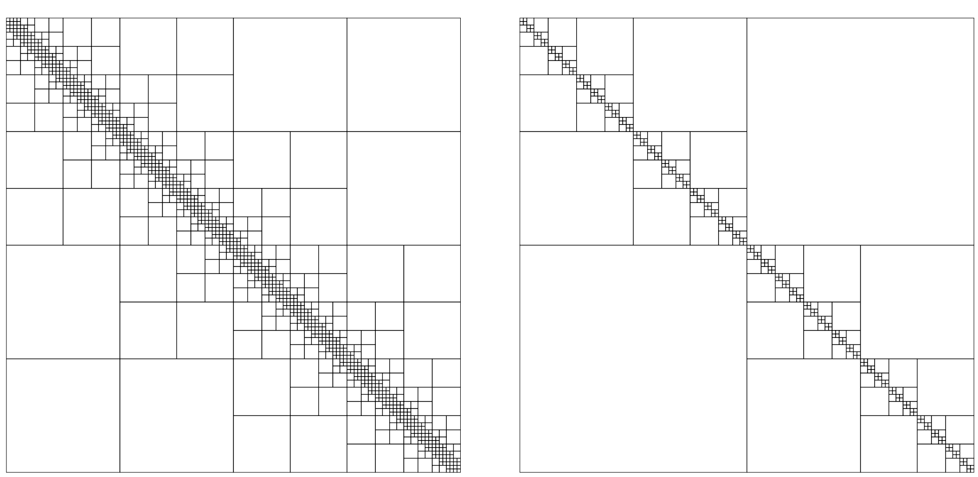


Figure 1: Standard- (left) and Weak-admissible $\mathcal{H}$-partitionings for $d = 1$.

Let S_N be the space of sequences $\{f[n]\}_{0\le n<N}$ of period N.

S_N is an Euclidean space, $\langle f,g\rangle = \sum\limits_{n=0}^{N-1} f[n]g^*[n]$.

Def. 2.2. The *discrete Fourier transform* (DFT) of f is

$$\widehat{f}[k] := \langle f, e_k\rangle = \sum_{n=0}^{N-1} f[n]\exp\left(\frac{-2i\pi kn}{N}\right), \quad (N^2 \text{ complex multiplications}).$$

The FT matrix $F_N = \{f_{k,n}\}_{k,n=1}^{N}$ is given by

$$f_{k,n} := \exp(\frac{-2i\pi kn}{N}) = W^{-nk}, \quad W = e^{2i\pi/N}.$$

The DFT(N) can be calculated by Fast Fourier Transform (FFT) in $\mathcal{N}_{FFT}(N) = C_F N \log_2 N$ operations, $C_F \approx 4$.

The FFT traces back (1805) to Gauss **(1777 - 1855)**.

First computer program **Coolly/Tukey (1965)**.

Let g be the discrete convolution of two signals f, h supported only by the indices $0 \le n \le M-1$,

$$g[n] = (f * h)[n] = \sum_{k=-\infty}^{\infty} f[k]h[n-k].$$

The naive implementation requires $M(M+1)$ operations.

It can be represented as a matrix-by-vector product (MVP) with the Toeplitz matrix

$$T = \{h[n-k]\}_{0\le n,k<M} \in \mathbb{R}^{M\times M}, \quad g = Tf.$$

Extending f and h with over M samples by

$$\tilde{h}[M] = 0, \quad \tilde{h}[2M-i] = h[i], \quad i = 1,...,M-1,$$

$$\tilde{f}[n] = 0, \quad n = M,...,2M-1,$$

we reduce the problem to the MVP with a circulant matrix $\mathcal{C} \in \mathbb{R}^{2M\times 2M}$ specified by the first row $\tilde{h} \in \mathbb{R}^{2M}$.

Huge problems: tensor methods bit super-computers

$\triangleright$ *The algebraic operations on high-dimensional data require heavy computing.*

$\triangleright$ *Linear cost* $O(N)$, $N = n^d$, *is satisfactory only for small* d.

$\triangleright$ *Traditional "asymptotically optimal" methods suffer from the* **"curse of dimensionality"**

$\triangleright$ *Complexity of matrix operations in full arithmetics:* $O(N^3)$.
It is too large already for $d = 3$, i.e., $N = n^3 \Rightarrow N^3 = n^9$.

$\triangleright$ *A paradigm* of up-to-date numerical simulations:

The higher computer capacities *do not relax the curse of dimensionality*.

$\triangleright$ **Remedy:** The identification and efficient use of low rank tensor structured representations with linear scaling in d.

Literature to Lecture 2(I)

1. G.H. Golub and C.F. Van Loan: Matrix computations. 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

2. W. Hackbusch: *Hierarchiche Matrizen - Algorithmen und Analysis.* Springer 2009.

3. M. Bebendorf: *Hierarchical Matrices.* Springer, 2008.

4. W. Hackbusch and B.N. Khoromskij: *A Sparse $\mathcal{H}$-matrix Arithmetic. Part II: Application to Multi-Dimensional Problems.* Copmputing 64 (2000), 21-47.

5. B.N. Khoromskij: *Data-Sparse Approximation of Integral Operators.* Lecture notes 17, MPI MIS, Leipzig 2003, 1-61.

6. E. Tyrtyshnikov: *Incomplete cross approximation in the mosaic-skeleton method.* Computing 64 (2000), 367-380.

http://personal-homepages.mis.mpg.de/bokh

Outline of Lecture 2(II).

1. Tensor product of finite dimensional Hilbert spaces (multidimensional vectors).
2. Matrix unfolding and contracted product of tensors.
3. Tensor rank and canonical representation.
4. Rank decomposition can be useful in linear algebra: $O(n^{\log_2 7})$- Strassen algorithm of matrix multiplication.
5. Orthogonal Tucker and mixed Tucker-canonical models.
6. Linear and multilinear operations on "formatted tensors".
7. Toward best (nonlinear) approx. in basic tensor formats.

Let $\mathbb{H} = H_1 \otimes ... \otimes H_d$ be a tensor prod. Hilbert space (TPHS).

H_ℓ is a real Euclidean space of vectors,

$$H_\ell = \mathbb{R}^{n_\ell}, \quad n_\ell \in \mathbb{N}, \quad n_\ell := \dim H_\ell, \quad \ell = 1, ..., d.$$

The scalar product of rank-1 elements $W, V \in \mathbb{H}$ is given by

$$\langle W, V \rangle = \langle w^{(1)} \otimes \ldots \otimes w^{(d)}, v^{(1)} \otimes \ldots \otimes v^{(d)} \rangle = \prod_{\ell=1}^{d} \langle w^{(\ell)}, v^{(\ell)} \rangle_{H_\ell}, \quad (2)$$

$$W(i_1, ..., i_d) = \prod_{\ell=1}^{d} w^{(\ell)}(i_\ell), \quad Stor(W) = n_1 + ... + n_d \ll \prod_{\ell=1}^{d} n_\ell.$$

Choose a basis $\left\{\phi_k^{(\ell)} : 1 \leq k \leq n_\ell\right\}$ of H_ℓ, then the set $\{\phi_{k_1}^{(1)} \otimes \phi_{k_2}^{(2)} \otimes \ldots \otimes \phi_{k_d}^{(d)}\}$ $(1 \leq k_\ell \leq n_\ell,\ 1 \leq \ell \leq d)$ is the basis in $\mathbb{H}$.

Denote the d-fold tensor prod. $\mathbb{H} = H \otimes ... \otimes H$ by $H^{\otimes d}\,(= \mathbb{R}^{I^d})$.

Tensor product of finite dimensional Hilbert spaces

Rem. 2.1. d-th order tensor $A \in \mathbb{H}$ of size $\mathbf{n} = (n_1, ..., n_d)$ is a function of d discrete arguments (multi-dimensional array/vector over $\mathcal{I} := I_1 \times ... \times I_d$, $I_\ell = \{1, ..., n_\ell\}$), i.e.,

$$A : I_1 \times ... \times I_d \to \mathbb{R}, \quad \text{with} \quad dim(\mathbb{H}) = |\mathbf{n}| = n_1 \cdots n_d.$$

Notations for the coordinate representation of A,

$$A := [a_{i_1 \dots i_d}] = [A(i_1, ..., i_d)] \in \mathbb{R}^{\mathcal{I}}.$$

The *Euclidean scalar product* of tensors $A, B \in \mathbb{H}$ becomes

$$\langle A, B \rangle := \sum_{(i_1, \dots, i_d) \in \mathcal{I}} a_{i_1 \dots i_d} b_{i_1 \dots i_d},$$

inducing the Euclidean (Frobenious) norm $\|A\|_F := \sqrt{\langle A, A \rangle}$.

The dimension directions $\ell = 1, ..., d$ are called the *modes*.

Tensor is a union of *ℓ-mode fibers*, $A(i_1, ..., i_{\ell-1}, :, i_{\ell+1}, ..., i_d)$.

Vectorization of a tensor

For a matrix $A \in \mathbb{R}^{m \times n}$ we use the *vector representation* (vectorization or concatenation) $A \to vec(A) \in \mathbb{R}^{mn}$, where $vec(A)$ is an $nm \times 1$ vector obtained by "stacking" A's columns (the FORTRAN-style ordering)

$$vec(A) := [a_{11}, ..., a_{n1}, a_{12}, ..., a_{nm}]^T.$$

In this way, $vec(A)$ is a rearranged version of A.

Def. 2.1. In general, if $A \in \mathbb{R}^{I_1 \times \dots \times I_d}$ is a tensor, then the vectorization of A is recursively defined by

$$vec(A) = \begin{bmatrix} vec([A(i_1, ..., i_{d-1}, 1)]) \\ vec([A(i_1, ..., i_{d-1}, 2)]) \\ \vdots \\ vec([A(i_1, ..., i_{d-1}, n_d)]) \end{bmatrix} \in \mathbb{R}^{|\mathbf{n}| \times 1}.$$

The tensor element $A(i_1, ..., i_d)$ maps to vector entry $(j, 1)$, where $j = 1 + \sum_{k=1}^{d} (i_k - 1) \prod_{\ell=1}^{k-1} n_\ell$.

Unfolding of a tensor into a matrix (matricization) is a way to map high order tensor into two-fold arrays by rearranging (reshaping) it for some $\ell \in \{1,...,d\}$, $\mathbb{R}^{\mathcal{I}} \mapsto \mathbb{R}^{I_\ell \times I_{(-\ell)}}$, and then vectorizing the tensors in $\mathbb{R}^{i_\ell \times I_{(-\ell)}}$ for each $i_\ell \in I_\ell$. The single hole index set is defined by $I_{(-\ell)} := I_1 \times ... \times I_{\ell-1} \times I_{\ell+1} \times ... \times I_d$.

Def. 2.2. The unfolding $mat(A)$ of a tensor $A \in \mathbb{R}^{I_1 \times ... \times I_d}$ w.r.t. the index ℓ (along mode ℓ) is defined by a matrix $mat(A) := A_{(\ell)}$ of dimension $n_\ell \times \bar{n}_\ell$, so that the tensor element $A(i_1,...,i_d)$ maps to matrix element $v(i_\ell, j)$, $i_\ell \in I_\ell$, where

$$A_{(\ell)} = [v_{i_\ell j}], \text{ with } \quad j \in \{1,\ldots,\bar{n}_\ell\},\ \bar{n}_\ell = n_1 \cdots n_{\ell-1} n_{\ell+1} \cdots n_d,$$

$$j = 1 + \sum_{k=1, k\neq \ell}^{d} (i_k - 1) J_k, \quad J_k = \prod_{m=1, m\neq\ell}^{k-1} n_m.$$

Exer. 2.2. ($mat(A)$ by recursion over $vec(A)$). Derive the representation

$$mat(A) = [vec([A(i_1,...,i_{\ell-1},1,i_{\ell+1},...,i_d)],...,vec([A(i_1,...,i_{\ell-1},n_\ell,i_{\ell+1},...,i_d)])]^T.$$

Rem. 2.2. Kolmogorow's decomposition is a particular way for unforlding of the multivariate function into "one-dimensional" representation (univariate function).

Ex. 2.1. Define a tensor $A \in \mathbb{R}^{3\times 2\times 3}$ by
$a_{111} = a_{112} = a_{211} = -a_{212} = 1$,
$a_{213} = a_{311} = a_{313} = a_{121} = a_{122} = a_{221} = -a_{222} = 2$,
$a_{223} = a_{321} = a_{323} = 4$, $a_{113} = a_{312} = a_{123} = a_{322} = 0$.

The matrix unfolding $A_{(1)}$ is given by

$$A_{(1)} = \begin{bmatrix} 1 & 1 & 0 & 2 & 2 & 0 \\ 1 & -1 & 2 & 2 & -2 & 4 \\ 2 & 0 & 2 & 4 & 0 & 4 \end{bmatrix}.$$

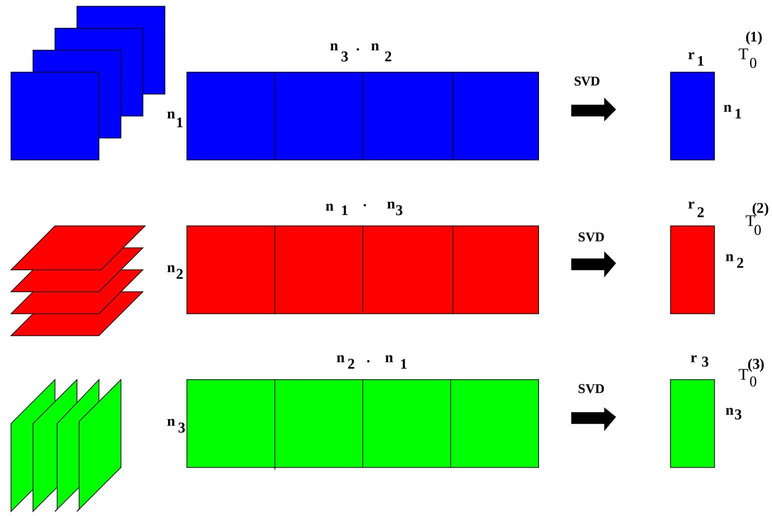


Figure 2: Visualization of the matrix unfolding for $d = 3$.

Def. 2.3. The ℓ-rank of A ($\ell = 1, ..., d$), denoted by $R_\ell = rank_\ell(A)$, is the dimension of the vector space spanned by the ℓ-mode vectors (fibers).

The ℓ-mode fibers of A are the column vectors of the matrix unfolding $A_{(\ell)}$ (by definition).

Prop. 2.1. We have

$$rank_\ell(A) = rank(A_{(\ell)}).$$

The major difference with the matrix case, however, is the fact that the different ℓ-ranks of a higher-order tensor are not necessarily the same.

An important tensor-tensor operation is the *contracted product* of two tensors, in particular, a *tensor-matrix contracted product* along mode ℓ.

Def. 2.4. Given $V \in \mathbb{R}^{I_1\times\ldots\times I_d}$, and a matrix $M \in \mathbb{R}^{J_\ell \times I_\ell}$, define the mode-$\ell$ tensor-matrix contracted product by

$$U = V \times_\ell M \in \mathbb{R}^{I_1\times\ldots\times I_{\ell-1}\times J_\ell \times I_{\ell+1}\ldots\times I_d},$$

where

$$u_{i_1,\ldots,i_{\ell-1},j_\ell,i_{\ell-1},\ldots,i_d} = \sum_{i_\ell=1}^{n_\ell} v_{i_1,\ldots,i_{\ell-1},i_\ell,i_{\ell-1},\ldots,i_d} m_{j_\ell,i_\ell}, \quad j_\ell \in J_\ell.$$

This is the generalization of the matrix-matrix multiplication:

$$M_{(n,m)} \times_2 M_{(p,m)} = M_{(n,m)} M^T_{(p,m)} \to M_{(n,p)}.$$

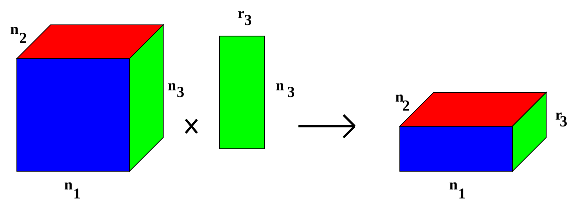


Figure 3: Contracted product of a third-order tensor with a matrix.

Rem. 2.3. A dth-order tensor A has rank 1, $rank(A) = 1$, if it is the contracted product of d vectors $t^{(1)}, \ldots, t^{(d)}$, $t^{(\ell)} \in \mathbb{R}^{I_\ell}$,

$$A = t^{(1)} \times_2 t^{(2)} \ldots \times_d t^{(d)}, \quad a_{i_1\ldots i_d} = t^{(1)}_{i_1} \ldots t^{(d)}_{i_d},$$

for $i_\ell \in I_\ell$ $(\ell = 1, \ldots, d)$.

Ex. 2.2. Let $A = a_1 \otimes a_2$, $B = b_1 \otimes b_2$, $a_i, b_i \in \mathbb{R}^n$ $(d = 2)$.

$$\langle A, B\rangle = \langle a_1, b_1\rangle\langle a_2, b_2\rangle, \quad ||A||_F = \sqrt{\langle a_1, a_1\rangle\langle a_2, a_2\rangle}.$$

Def. 2.5. (Canonical (CP) format). Choose a subset of those elements which require only R terms,

$$\mathcal{C}_R = \left\{ w \in \mathbb{H} : w = \sum_{k=1}^{R} w_k^{(1)} \otimes w_k^{(2)} \otimes \ldots \otimes w_k^{(d)}, \ w_k^{(\ell)} \in H_\ell \right\}.$$

Elem. $w \in \mathcal{C}_R$, $w \notin \mathcal{C}_{R-1}$, are called to have the *tensor rank* R.

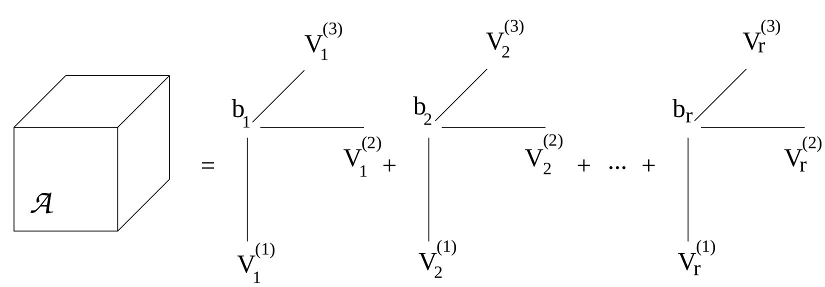
Pro and contra for canonical format

Tensors $w \in \mathcal{C}_R$ can be represented by the description of Rd elements $w_k^{(\ell)} \in H_\ell$, i.e. with linear cost in d, dRn.

Advantages: Tremendous reduction of storage cost, removing d from the exponential, $n^d \to d\,R\,n$;
Analytic methods of low-rank approx. for Green's kernels.

Limitations: $\mathcal{C}_R$ is a nonclosed set. Approximation process in $\mathcal{C}_R$ is not robust. Exact rank-R represent. is N-P hard.

Visualization of the canonical model for $d = 3$.



Strassen algorithm via rank decomposition

Finding the tensor rank can be a useful concept even in the classical linear algebra.

Historical remarks on the Strassen algorithm of fast matrix-matrix multiplication of complexity $O(n^{\log_2 7})$.

$O(n^{2+\varepsilon})$ algorithm to multiply two $n \times n$ matrices gives $O(n^{2+\varepsilon})$ method for solving system of n linear eqs. [Strassen 1969].

Best known result: $O(n^{2.376})$ [Copperesmith-Winograd 1987].

Lloyd N. Trefethen bets Peter Alfred (25 June 1985) that a method will have been found to solve $Ax = b$ in $O(n^{2+\varepsilon})$ operations for any $\varepsilon > 0$ (numerical stabiliy is not an isue).

Details at personal homepage by Prof. L.N. Trefethen (Uni. Oxford).

Strassen algorithm via rank decomposition

In the block form

$$\begin{bmatrix} C_1\, C_2 \\ C_3\, C_4 \end{bmatrix} = \begin{bmatrix} A_1\, A_2 \\ A_3\, A_4 \end{bmatrix} \cdot \begin{bmatrix} B_1\, B_2 \\ B_3\, B_4 \end{bmatrix}$$

with

$$C_k = \sum_{i=1}^{4}\sum_{j=1}^{4} \gamma_{ijk} A_i B_j, \quad k = 1, ..., 4,$$

where for the 3-rd order coefficients tensor of size $4 \times 4 \times 4$ we have (slicewise)

$$\{\gamma_{ijk}\} = \triangleleft_1 \begin{bmatrix} 1000 \\ 0010 \\ 0000 \\ 0000 \end{bmatrix} \triangleleft_2 \begin{bmatrix} 0100 \\ 0001 \\ 0000 \\ 0000 \end{bmatrix} \triangleleft_3 \begin{bmatrix} 0000 \\ 0000 \\ 1000 \\ 0010 \end{bmatrix} \triangleleft_4 \begin{bmatrix} 0000 \\ 0000 \\ 0100 \\ 0001 \end{bmatrix}.$$

Here $\triangleleft_i$ means that the related matrix corresponds to slice number $i \leq 4$.

Strassen algorithm via rank decomposition

Suppose that we have rank-R expansion

$$\gamma_{ijk} = \sum_{t=1}^{R} u_{it} v_{jt} w_{kt}.$$

Then

$$C_k = \sum_{t=1}^{R} w_{kt} \sum_{i=1}^{4}\sum_{j=1}^{4} u_{it} A_i v_{jt} B_j = \sum_{t=1}^{R} w_{kt} \left(\sum_{i=1}^{4} u_{it} A_i\right)\left(\sum_{j=1}^{4} v_{jt} B_j\right).$$

Precompute $\Sigma_t = \sum_{i=1}^{4} u_{it} A_i$, $\Delta_t = \sum_{j=1}^{4} v_{jt} B_j$ and reduce the initial task to R matrix-matrix products of size $n/2 \times n/2$.

We have $R \leq 8$ (**why ?**), but there are representations (infinitely many) of rank 7 (Strassen's result).

Open problem: Is it possible to construct rank decompositions with $R < 7$? If yes, then the Strassen result can be improved.

Exer. 2.3. Try to compute the canonical rank-7 decomposition of γ by the Tensor Toolbox.

As in the Galerkin method, the replacement of H_ℓ by subspaces $V_\ell \subset H_\ell$ $(1 \le \ell \le d)$ leads to the tensor subspace

$$\mathbb{V} = V_1 \otimes V_2 \otimes \ldots \otimes V_d \subset \mathbb{H}.$$

Setting $r_\ell := \dim V_\ell$ and choosing a orthonormal basis $\left\{\phi_k^{(\ell)} : 1 \le k \le r_\ell\right\}$ of V_ℓ, we can represent each $v \in \mathbb{V}$ by

$$v = \sum_{\mathbf{k}} b_{\mathbf{k}} \phi_{k_1}^{(1)} \otimes \phi_{k_2}^{(2)} \otimes \ldots \otimes \phi_{k_d}^{(d)}, \text{ with } b_{\mathbf{k}} \in \mathbb{R}^{J_1 \times \ldots \times J_d},$$

and with the multi-index $\mathbf{k} = (k_1, \ldots, k_d)$, $1 \le k_\ell \le r_\ell$, where $J_\ell := \{1, ..., r_\ell\}$, $(1 \le \ell \le d)$.

Let $\mathbf{r} = (r_1, \ldots, r_d) \in \mathbb{N}^d$ be a d-tuple of dimensions.

Exer. 2.4. Max. *canonical rank* in $\mathbb{V}$, $R = (\prod_{\ell=1}^d r_\ell)/\max_\ell r_\ell$.

Def 2.6. (Tucker format) Given $\mathbf{r}$, define

$$\mathcal{T}_{\mathbf{r}} := \{v \in \mathbb{V} \subset \mathbb{H} \quad \forall\, V_\ell \text{ s.t. } \dim V_\ell = r_\ell, \quad \ell = 1, ..., d\}.$$

A representation of $w \in \mathcal{T}_{\mathbf{r}}$ is called a Tucker format of rank $\mathbf{r}$ (cf. [1], [3], [4]).

Denote by $U^{(\ell)} = [\phi_1^{(\ell)}, ..., \phi_{r_\ell}^{(\ell)}] \in \mathbb{R}^{n_\ell \times r_\ell}$ the ℓ-mode side matrix.

Def. 2.7. We say that $U^{(\ell)} \in \mathbb{S}_{r_\ell}$, where $\mathbb{S}_{r_\ell}$ is the Stiefel manifold of the orthogonal $n_\ell \times r_\ell$ matrices.

The Tucker representation is not unique (rotation of $U^{(\ell)}$).

Let us set for ease of presentation, $n = n_\ell$, $(\ell = 1, ..., d)$.

Storage of $w \in \mathcal{T}_{\mathbf{r}}$: $\prod_{\ell=1}^d r_\ell$ reals and the sampling of $\sum_{\ell=1}^d r_\ell$ vectors $\phi_k^{(\ell)} \in \mathbb{R}^n$, $O(r^d + drn)$, $r = \max r_\ell$ (curse of dimension).

Comment to Def. 2.6. Using the (orthogonal) side-matrices

$$U^{(\ell)} = [\phi_1^{(\ell)} ... \phi_{r_\ell}^{(\ell)}] \in \mathbb{R}^{n \times r_\ell},$$

we represent the Tucker decomposition of $V \in \boldsymbol{\mathcal{T}}_{\mathbf{r}}$ as a tensor-by-matrix contracted products,

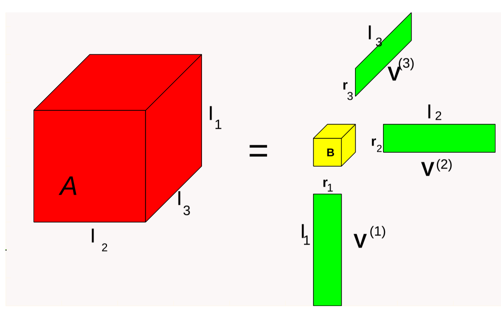
$$V = \boldsymbol{\beta} \times_1 U^{(1)} \times_2 U^{(2)} ... \times_d U^{(d)},$$

where $\boldsymbol{\beta} \in \mathbb{R}^{J_1 \times ... \times J_d}$ is the core tensor of "small" size $r_1 \times ... \times r_d$.

Rem. 2.4. In the case $d = 2$, the above representation is a multilinear equivalent of a matrix factorisation, i.e., we have

$$A = \boldsymbol{\beta} \times_1 U^{(1)} \times_2 U^{(2)} = U^{(1)} \cdot \boldsymbol{\beta} \cdot {U^{(2)}}^T, \quad \boldsymbol{\beta} \in \mathbb{R}^{r_1 \times r_2}.$$

Visualization of the Tucker model for $d = 3$:



How to relax drowbacks of both $\boldsymbol{\mathcal{T}}_{\mathbf{r},\mathbf{n}}$ and $\boldsymbol{\mathcal{C}}_R$?

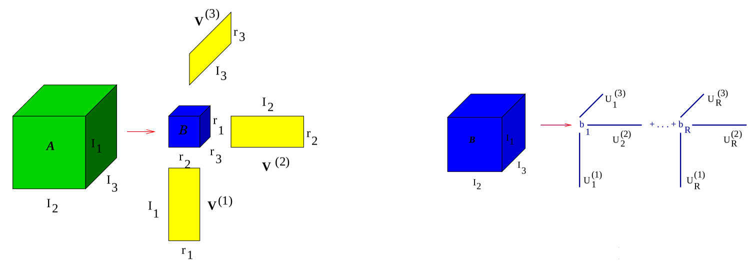
Main idea: The two-level tensor format that inherits the Tucker orthogonality in *primal space* (robust decomposition) and the $\boldsymbol{\mathcal{C}}_R$ structure in the *dual (coefficients) space* (linear scaling in d, n, R, r).

Def. 2.8. *Mixed Tucker-canonical model* ($\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}$), ([2]).

Given the rank parameters $\mathbf{r}, R$ (normally, $r \ll R$), define a subclass $\boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}} \subset \boldsymbol{\mathcal{T}}_{\mathbf{r},\mathbf{n}}$ of tensors with $\boldsymbol{\beta} \in \boldsymbol{\mathcal{C}}_{R,\mathbf{r}} \subset \mathbb{R}^{J_1\times\ldots\times J_d}$,

$$V = \left(\sum_{\nu=1}^{R} \beta_\nu u_\nu^{(1)} \otimes \ldots \otimes u_\nu^{(d)}\right) \times_1 V^{(1)} \times_2 V^{(2)} \ldots \times_d V^{(d)}.$$

Storage: $S(V) = dRr + R + drn$ (linear scaling in d, n, R, r).



Level I: Tucker decomposition (left). Level II: canonical decomposition of $\boldsymbol{\beta}$ (right).

Exer. 2.5. Compute the mixed decomposition of functional tensor for $f_{1,\kappa}$, is it much faster than CP? (cf. Lect. 1).

Exer. 2.6. Compute the canonical, Tucker and ℓ-mode ε-rank of the Hilbert tensor $A = \{a_{ijk}\}$, $a_{ijk} = 1/(i+j+k)$ $(i,j,k = 1,...,n)$ with $n = 10^2$, corresponding to approximation error $\varepsilon = 10^{-3}, 10^{-4}, 10^{-5}$. Do you observe the exponential convergence in r_ε? (Hint: See Exer. 2.1)

Probl. 1. Efficient and accurate MLA in fixed tensor classes $\mathcal{S}$ getting rid of the curse of dimensionality.

Probl. 2. Best rank-structured approximation of a high-order tensor $f \in \mathbb{V}_\mathbf{n}$ in the fixed set $\mathcal{S} \subset \{\mathcal{T}_\mathbf{r}, \mathcal{C}_R, \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}\}$.

Probl. 3. For fixed accuracy $\varepsilon > 0$, efficient approximation of a high-order tensor $f \in \mathbb{V}_\mathbf{n}$ in $\mathcal{S}$ with adaptive rank parameter.

Since both $\mathcal{T}_\mathbf{r}$ and $\mathcal{C}_R$ are not linear spaces, we arrive at a nontrivial nonlinear approximation problem on estimation:

Given $X \in \mathbb{V}_\mathbf{n}$ (more generally, $X \in \mathcal{S}_0 \subset \mathbb{V}_\mathbf{n}$), find

$$T_\mathbf{r}(X) := \underset{A\in\mathcal{S}}{\operatorname{argmin}} \|X - A\|, \quad \text{where} \quad \mathcal{S} \subset \{\mathcal{T}_\mathbf{r}, \mathcal{C}_R, \boldsymbol{\mathcal{T}}_{\mathcal{C}_{R,\mathbf{r}}}\}. \qquad (3)$$

Recall that the decomposition

$$f(x) := \sin(\sum_{j=1}^{d} x_j) = \sum_{j=1}^{d} \sin(x_j) \prod_{k\in\{1,\dots,d\}\setminus\{j\}} \frac{\sin(x_k+\alpha_k-\alpha_j)}{\sin(\alpha_k-\alpha_j)} \quad (4)$$

holds for any $\alpha_k \in \mathbb{R}$, s.t. $\sin(\alpha_k - \alpha_j) \neq 0$ for all $j \neq k$.

(4) shows the lack of uniqueness (ambiguity) of the "best" rank-d tensor representation. The convergence of ALS schemes in $\mathcal{C}_R$ might be non-robust (multiple local minima).

Exer. 2.7. Prove that the tensor related to $f(x)$ has the maximal Tucker rank 2. Check it by Tensor Toolbox.

Principal discussion: **How to solve (3) efficiently?**

Main aproaches: MLA on formatted tensors + high-order extension(s) of trunc. SVD + nonlinear iteration + multigrid.

1. L. De Lathauwer, B. De Moor, J. Vandewalle: *On the best rank-1 and rank-*$(R_1, \dots, R_N)$ *approximation of higher-order tensors.* SIAM J. Matrix Anal. Appl., **21** (2000) 1324-1342.

2. B.N. Khoromskij: Structured Rank-$(r_1, \dots, r_d)$ Decomposition of Function-related Tensors in $\mathbb{R}^d$. Comp. Meth. in Appl. Math., V. 6 (2006), 194-220.

3. B.N. Khoromskij and V. Khoromskaia, *Multigrid Tensor Approximation of Function Related Arrays.* SIAM J. on Sci. Comp., **31**(4), 3002-3026 (2009).

4. T.G. Kolda, and B.W. Bader: *Tensor decompositions and applications.* SIAM Review, 51/3 (2009), 455-500.

5. I.Oseledets, D. Savostyanov, E.Tyrtyshnikov, *Linear algebra for tensor problems*, Computing, 85 (2009), 169–188.

6. L.R. Tucker: Some mathematical notes on three-mode factor analysis. Psychometrika 31 (1966) 279-311.

http://personal-homepages.mis.mpg.de/bokh