# NUMERICAL METHODS WITH TENSOR REPRESENTATIONS OF DATA

Eugene Tyrtyshnikov

Institute of Numerical Mathematics of Russian Academy of Sciences

eugene.tyrtyshnikov@gmail.com

2 June 2012

# COLLABORATION

MOSCOW:

I.Oseledets, D.Savostyanov
S.Dolgov, V.Kazeev, O.Lebedeva,
A.Setukha, S.Stavtsev, D.Zheltkov
S.Goreinov, N.Zamarashkin

LEIPZIG:

W.Hackbusch, B.Khoromskij, R.Schneider
H.-J.Flad, V.Khoromskaia,
M.Espig, L.Grasedyck

# NUMERICAL METHODS WITH TENSORIZATION OF DATA

We consider typical problems of numerical analysis (matrix computations, interpolation, optimization) under the assumption that the input, output and all intermediate data are represented by
*tensors with many dimensions*
(tens, hundreds, even thousands).

Of course, it assumes a very special structure of data. But we have it in really many problems!

# THE CURSE OF DIMENSIONALITY

The main problem is that *using arrays* as means to introduce tensors in many dimensions is *infeasible*:

- if $d = 300$ and $n = 2$, then such an array contains $2^{300} \gg 10^{83}$ entries

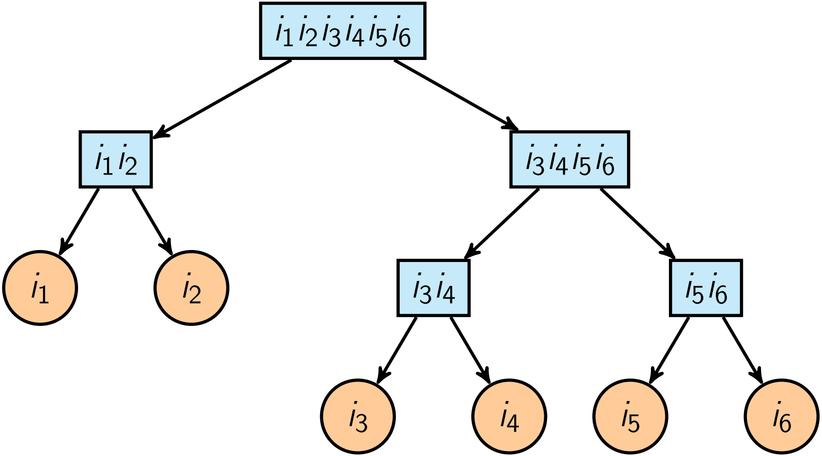
# NEW REPRESENTATION FORMATS

Canonical polyadic and Tucker decompositions are of limited use for our purposes (by different reasons).

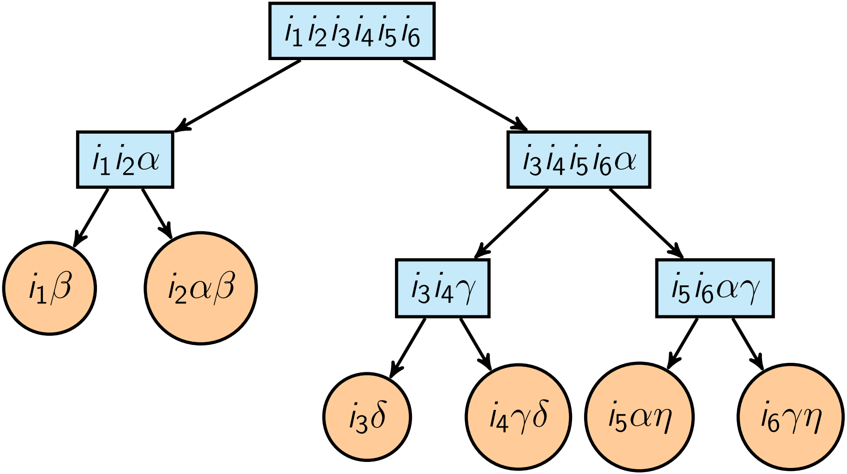
New decompositions:

- TT (Tensor Train)
- HT (Hierarchical Tucker)

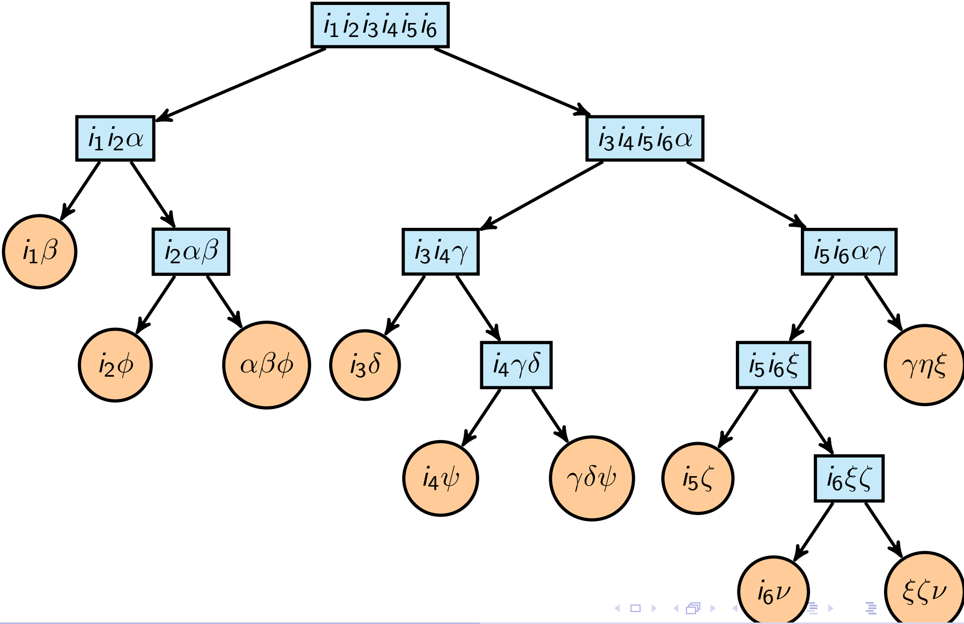
# REDUCTION OF DIMENSIONALITY



# SCHEME FOR TT



# SCHEME FOR HT



# THE BLESSING OF DIMENSIONALITY

TT and HT provide new *representation formats* for $d$-tensors $+$ algorithms with *complexity linear in d*.

Let the amount of data be $N$. In numerical analysis, complexity $O(N)$ is usually considered as a dream.

With ultimate tensorization we go *beyond the dream*:

since $d \sim \log N$, we may obtain *complexity* $O(\log N)$.

# BASIC TT ALGORITHMS

- `TT rounding.`
  Like the rounding of machine numbers.
  $\mathtt{COMLEXITY} = O(dnr^3)$.
  $\mathtt{ERROR} \leqslant \sqrt{d-1} \cdot \mathtt{BEST\ ERROR}$.

- `TT interpolation.`
  A tensor train is constructed from sufficiently few elements of the tensor, the number of them is $O(dnr^2)$.

- `TT quantization and wavelets.`
  Low-dimensional $\to$ high-dimensional $\Rightarrow$ algebraic wavelet tranbsforms (WTT).

  In matrix problems the complexity may drop from $O(N)$ down to $O(\log N)$.

# SUMMATION AGREEMENT

Omit the symbol of summation. Assume summation if the index in a product of quantities with indices is repeated at least twice. Equations hold for all values of other indices.

# SKELETON DECOMPOSITION

$$A = UV^{\top} = \sum_{\alpha=1}^{r} \begin{bmatrix} u_{1\alpha} \\ \dots \\ u_{m\alpha} \end{bmatrix} \begin{bmatrix} v_{1\alpha} & \dots & v_{n\alpha} \end{bmatrix}$$

According to the summation agreement,

$$a(i,j) = u(i,\alpha)v(j,\alpha)$$

# CANONICAL AND TUCKER

## CANONICAL DECOMPOSITION

$$a(i_1 \dots i_d) = u_1(i_1\alpha) \dots u_d(i_d\alpha)$$

## TUCKER DECOMPOSITION

$$a(i_1 \dots i_d) = g(\alpha_1 \dots \alpha_d) u_1(i_1\alpha_1) \dots u_d(i_d\alpha_d)$$

# TENSOR TRAIN (TT) IN THREE DIMENSIONS

$$a(i_1 ;\ i_2 i_3) = g_1(i_1 ;\ \alpha_1) a_1(\alpha_1 ;\ i_2 i_3)$$

$$a_1(\alpha_1 i_2 ;\ i_3) = g_2(\alpha_1 i_2 ;\ \alpha_2) g_3(\alpha_2 ;\ i_3)$$

## TENSOR TRAIN (TT)

$$a(i_1 i_2 i_3) = g_1(i_1 \alpha_1) g_2(\alpha_1 i_2 \alpha_2) g_3(\alpha_2 i_3)$$

# TENSOR TRAIN (TT) IN $d$ DIMENSIONS

$$a(i_1 \dots i_d) =$$

$$g_1(i_1\alpha_1)g_2(\alpha_1 i_2 \alpha_2)\dots$$
$$g_{d-1}(\alpha_{d-2} i_{d-1} \alpha_{d-1}) g_d(\alpha_{d-1} i_d)$$

$$a(i_1 \dots i_d) = \prod_{k=1}^{d} g_k(\alpha_{k-1} i_k \alpha_k)$$

# KRONECKER REPRESENTATION OF TENSOR TRAINS

$$A = G^1_{\alpha_1} \otimes G^2_{\alpha_1\alpha_2} \otimes \ldots \otimes G^{d-1}_{\alpha_{d-2}\alpha_{d-1}} \otimes G^d_{\alpha_{d-1}}$$

$A$ is of size $(m_1 \ldots m_d) \times (n_1 \ldots n_d)$.

$G^k_{\alpha_{k-1}\alpha_k}$ is of size $m_k \times n_k$.

# ADVANTAGES OF TENSOR-TRAIN REPRESENTATION

The tensor is determined through
$d$ *tensor carriages* $g_k(\alpha_{k-1} i_k \alpha_k)$,
each of size $r_{k-1} \times n_k \times r_k$.

If the maximal size is $r \times n \times r$,
then the number of representation parameters
does not exceed $dnr^2 \ll n^d$.

# TENSOR TRAIN PROVIDES STRUCTURED SKELETON DECOMPOSITIONS OF UNFOLDING MATRICES

$$A_k = a(i_1 \dots i_k \,;\, i_{k+1} \dots i_d) =$$

$$u_k(i_1 \dots i_k \,;\, \alpha_k)\; v_k(\alpha_k \,;\, i_{k+1} \dots i_d) = U_k V_k^\top$$

$$u_k(i_1 \dots i_k \alpha_k) = g_1(i_1 \alpha_1) \dots g_k(\alpha_{k-1} i_k \alpha_k)$$

$$v_k(\alpha_k i_{k+1} \dots i_d) = g_{k+1}(\alpha_k i_{k+1} \alpha_{k+1}) \dots g_d(\alpha_{k-1} i_d)$$

# TT RANKS ARE BOUNDED BY THE RANKS OF UNFOLDING MATRICES

$$r_k \geqslant \mathrm{rank} A_k, \quad A_k = [a(i_1 \dots i_k \,;\, i_{k+1} \dots i_d)]$$

Equalities are always possible.

# ORTHOGONAL TENSOR CARRIAGES

A tensor carriage $g(\alpha i \beta)$ is called *row orthogonal* if its first unfolding matrix $g(\alpha\,;\, i\beta)$ has orthonormal rows.

A tensor carriage $g(\alpha i \beta)$ is called *column orthogonal* if its second unfolding matrix $g(\alpha i\,;\, \beta)$ has orthonormal columns.

# ORTHOGONALIZATION OF TENSOR CARRIAGES

$\forall$ tensor carriage $g(\alpha i \beta)$ $\exists$ decomposition

$$g(\alpha i \beta) = h(\alpha \alpha') q(\alpha' i \beta)$$

with $q(\alpha' i \beta)$ being row orthogonal.

$\forall$ tensor carriage $g(\alpha i \beta)$ $\exists$ decomposition

$$g(\alpha i \beta) = q(\alpha i \beta') h(\beta' \beta)$$

with $q(\alpha i \beta')$ being column orthogonal.

# PRODUCTS OF ORTHOGONAL TENSOR CARRIAGES

A product of row (column) orthogonal tensor carriages

$$p(\alpha_s, i_s \dots i_t, \alpha_t) = \prod_{k=s+1}^{t} g_k(\alpha_{k-1} i_k \alpha_k)$$

is also row (column) orthogonal.

# MAKING ALL CARRIAGES ORTHOGONAL

Orthogonalize the columns of $g_1 = q_1 h_1$, then compute and orthogonalize $h_1 g_2 = q_2 h_2$. Thus,

$$g_1 g_2 = q_1 q_2 h_2$$

and after $k$ steps

$$g_1 \dots g_k = q_1 \dots q_k h_k.$$

Similarly for the row orhogonalization,

$$g_{k+1} \dots g_d = h_{k+1} z_{k+1} \dots z_d.$$

# STRUCTURED ORTHOGONALIZATION

$\forall$ TT decomposition $a(i_1 \dots i_d) = \prod_{s=1}^{d} g_s(\alpha_{s-1} i_s \alpha_s)$

$\exists$ column $q_k$ and row $z_k$ orthogonal carriages s. t.

$$a(i_1 \dots i_k \,;\, i_{k+1} \dots i_d) =$$
$$\left( \prod_{s=1}^{k} q_k(\alpha'_{s-1} i_s \alpha'_s) \right) H_k(\alpha'_k, \alpha''_k) \left( \prod_{s=k+1}^{d} z_s(\alpha''_{s-1} i_s \alpha''_s) \right)$$

$q_k$ and $z_k$ can be constructed in $dnr^3$ operations.

# CONSEQUENCE: STRUCTURED SVD FOR ALL UNFOLDING MATRICES IN $O(dnr^3)$ OPERATIONS

It suffices to compute SVD for the matrices $H_k(\alpha'_k \alpha''_k)$.

# TENSOR APPROXIMATION VIA MATRIX APPROXIMATION

We can approximate any fixed unfolding matrix using its structured SVD:

$$a(i_1 \dots i_k \,;\, i_{k+1} \dots i_d) = a_k + e_k$$

$$a_k = U_k(i_1 \dots i_k \,;\, \alpha_k')\sigma_k(\alpha_k')V_k(\alpha_k' \,;\, i_{k+1} \dots i_d)$$

$$e_k = e_k(i_1 \dots i_k \,;\, i_{k+1} \dots i_d)$$

# ERROR ORTHOGONALITY

$$U_k(i_1 \dots i_k \alpha'_k) e_k(i_1 \dots i_k \,;\, i_{k+1} \dots i_d) = 0$$

$$e_k(i_1 \dots i_{k+1} \,;\, i_{k+1} \dots i_d) V_k(\alpha'_k i_{k+1} \dots i_d) = 0$$

# COROLLARY OF ERROR ORTHOGONALITY

Let $a_k$ be further approximated by a TT but so that $u_k$ or $v_k$ are kept. Then the further error, say $e_l$, is orthogonal to $e_k$. Hence,

$$||e_k + e_l||_F^2 = ||e_k||_F^2 + ||e_l||_F^2$$

# TENSOR-TRAIN ROUNDING

Approximate successively $A_1, A_2, \ldots, A_{d-1}$ with the error bound $\varepsilon$. Then

$$\texttt{FINAL ERROR} \ \leqslant \ \sqrt{d-1}\,\varepsilon$$

# TENSOR INTERPOLATION

Interpolate an implicitly given tensor by a TT using only *small part* of its elements, of order $dnr^2$.

Cross interpolation method for tensors is constructed as a generalization of the cross method for matrices (1995) and relies on the *maximal volume principle* from the matrix theory.

# MAXIMAL VOLUME PRINCIPLE

**THEOREM** (Goreinov, Tyrtyshnikov) *Let*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

*where $A_{11}$ is a $r \times r$ block with maximal determinant in modulus (volume) among all $r \times r$ blocks in $A$. Then the rank-$r$ matrix*

$$A_r = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} A_{11}^{-1} \begin{bmatrix} A_{11} & A_{12} \end{bmatrix}$$

*approximates $A$ with the Chebyshev-norm error at most in $(r+1)^2$ times larger than the error of best approximation of rank $r$.*

# BEST IS AN ENEMY OF GOOD

Move a *good submatrix* $M$ in $A$ to the upper $r \times r$ block. Use right-side multiplications by nonsingular matrices.

$$A = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ a_{r+1,1} & \dots & a_{r+1,r} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nr} \end{bmatrix}$$

NECESSARY FOR MAXIMAL VOLUME:

$$|a_{ij}| \leqslant 1, \quad r+1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant r$$

## COROLLARY OF MAXIMAL VOLUME

$$\sigma_{\min}(M) \geqslant 1/\sqrt{r(n-r)+1}$$

## ALGORITHM

- If $|a_{ij}| \geqslant 1+\delta$, then swap rows $i$ and $j$.
- Make identity matrix in the first $r$ rows by right-side multiplication.
- Quit if $|a_{ij}| < 1+\delta$ for all $i, j$. Otherwise repeat.

# MATRIX CROSS ALGORITHM

- Given *initial* column indices $j_1, \ldots, j_r$.
- Find *good* row indices $i_1, \ldots, i_r$ in these columns.
- Find *good* column indices in the rows $i_1, \ldots, i_r$.
- Proceed choosing good columns and rows until the skeleton cross approximations stabilize.

E.E.Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method, *Computing* 64, no. 4 (2000), 367–380.

# CROSS TENSOR-TRAIN INTERPOLATION

Let $a_1 = a(i_1, i_2, i_3, i_4)$. Seek crosses in the unfolding matrices.
On input: $r$ initial columns in each. Select *good* rows.

$$A_1 = [a(i_1\,;\; i_2, i_3, i_4)], \quad J_1 = \{i_2^{(\beta_1)} i_3^{(\beta_1)} i_4^{(\beta_1)}\}$$

$$A_2 = [a(i_1, i_2\,;\; i_3, i_4)], \quad J_2 = \{i_3^{(\beta_2)} i_4^{(\beta_2)}\}$$

$$A_3 = [a(i_1, i_2, i_3\,;\; i_4)], \quad J_3 = \{i_4^{(\beta_3)}\}$$

| rows | matrix | skeleton decomposition |
|---|---|---|
| $I_1 = \{i_1^{(\alpha_1)}\}$ | $a_1(i_1\,;\; i_2, i_3, i_4)$ | $a_1 = \sum_{\alpha_1} g_1(i_1; \alpha_1)\; a_2(\alpha_1;\; i_2, i_3, i_4)$ |
| $I_2 = \{i_1^{(\alpha_2)} i_2^{(\alpha_2)}\}$ | $a_2(\alpha_1, i_2\,;\; i_3, i_4)$ | $a_2 = \sum_{\alpha_2} g_2(\alpha_1, i_2;\; \alpha_2)\; a_3(\alpha_2, i_3;\; i_4)$ |
| $I_3 = \{i_1^{(\alpha_3)} i_2^{(\alpha_3)} i_3^{(\alpha_3)}\}$ | $a_3(\alpha_2, i_3\,;\; i_4)$ | $a_3 = \sum_{\alpha_3} g_3(\alpha_2, i_3;\; \alpha_3)\; g_4(\alpha_3;\; i_4)$ |

Finally

$$a = \sum_{\alpha_1, \alpha_2, \alpha_3, \alpha_4} g_1(i_1, \alpha_1)\; g_2(\alpha_1, i_2, \alpha_2)\; g_3(\alpha_2, i_3, \alpha_3)\; g_4(\alpha_3, i_4)$$

# QUANTIZATION OF DIMENSIONS

Increase the number of dimensions.

E.g. $2 \times \ldots \times 2$.

Extreme case is conversion of a vector of size $N = 2^d$ to a $d$-tensor of size $2 \times 2 \times \ldots \times 2$.

Using TT format with bounded TT ranks may reduce the complexity from $O(N)$ to as little as $O(\log_2 N)$.

# EXAMPLES OF QUANTIZATION

$f(x)$ is a function on $[0,1]$

$$a(i_1,\dots,i_d) = f(ih), \quad i = \frac{i_1}{2} + \frac{i_2}{2^2} + \dots + \frac{i_d}{2^d}$$

The array of values of $f$ is viewed as a tensor of size $2 \times \dots \times 2$.

```
EXAMPLE 1.
```
$f(x) = e^x + e^{2x} + e^{3x}$
```
ttrank= 2.7      ERROR=1.5e-14
```

```
EXAMPLE 2.
```
$f(x) = 1 + x + x^2 + x^3$
```
ttrank= 3.4      ERROR=2.4e-14
```

```
EXAMPLE 3.
```
$f(x) = 1/(x - 0.1)$
```
ttrank= 10.1     ERROR=5.4e-14
```

# THEOREMS

If there is an $\varepsilon$-approximation with separated variables

$$f(x+y) \approx \sum_{k=1}^{r} u_k(x) v_k(y), \quad r = r(\varepsilon),$$

then a TT exists with error $\varepsilon$ and TT-ranks $\leqslant r$.

If $f(x)$ is a sum of $r$ exponents, then an exact TT exists with the ranks $r$.

For a polynomial of degree $m$ an exact TT exists with the ranks $r = m+1$.

If $f(x) = 1/(x-\delta)$ then $r = \log \varepsilon^{-1} + \log \delta^{-1}$.

# ALGEBRAIC WAVELET FILTERS

$$a(i_1 \dots i_d) = u_1(i_1 \alpha_1) a_1(\alpha_1 i_2 \dots i_d) + e_1$$

$$u_1(i_1 \alpha_1) u(i_1 \alpha_1') = \delta(\alpha_1, \alpha_1')$$

$$a \longrightarrow a_1 = u_1 a \longrightarrow a_2 = u_2 a_1 \longrightarrow a_3 = u_3 a_2 \quad \dots$$

# TT QUADRATURE

$$I(d) = \int_{[0,1]^d} \sin(x_1 + x_2 + \ldots + x_d)\, dx_1 dx_2 \ldots dx_d =$$

$$\mathrm{Im} \int_{[0,1]^d} e^{i(x_1+x_2+\ldots+x_d)}\, dx_1 dx_2 \ldots dx_d = \mathrm{Im} \left( \left( \frac{e^i - 1}{i} \right)^d \right)$$

$n$ nodes in each dimension $\Rightarrow$ $n^d$ values in need!

TT interpolation method uses only *small part* ($n = 11$)

| $d$ | $I(d)$ | Relative Error | Timing |
|---|---|---|---|
| 500 | -7.287664e-10 | 2.370536e-12 | 4.64 |
| 1000 | -2.637513e-19 | 3.482065e-11 | 11.60 |
| 2000 | 2.628834e-37 | 8.905594e-12 | 33.05 |
| 4000 | 9.400335e-74 | 2.284085e-10 | 105.49 |

# QTT QUADRATURE

$$\int_0^\infty \frac{\sin x}{x} dx = \frac{\pi}{2}$$

Truncate the domain and use the rule of rectangles.

Machine accuracy causes to use $2^{77}$ values.
The vector of values is treated as a tensor of size
$2 \times 2 \times \ldots \times 2$.

TT-ranks $\leqslant 12$ for the machine precision.
Less than 1 sec on notebook.

# TT IN QUANTUM CHEMISTRY

Really many dimensions are natural in quantum molecular dynamics:

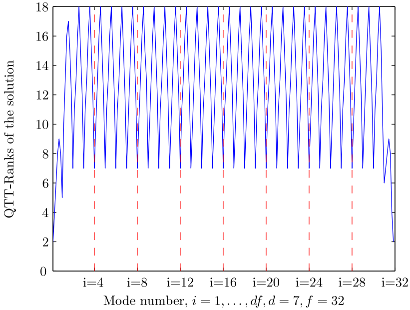
$$H\Psi = (-\frac{1}{2}\Delta + V(R_1, \dots, R_f))\Psi = E\Psi$$

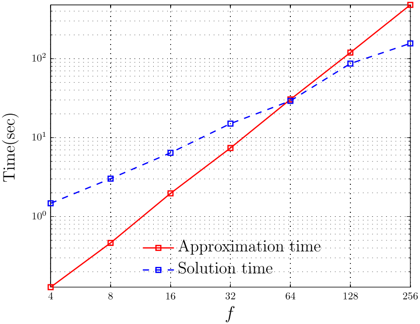
$V$ is a Potential Energy Surface (PES)

Calculation of $V$ requires to solve Schredinger equation for a variety of coordinates of atoms $R_1, \dots, R_f$. TT interpolation method uses only small part of values of $V$ from which it produces a suitable TT approximation of PES.

Henon-Heiles PES:

$$V(q_1, \ldots, q_f) = \frac{1}{2} \sum_{k=1}^{f} q_k^2 + \lambda \sum_{k=1}^{f-1} \left( q_k^2 q_{k+1} - \frac{1}{3} q_k^3 \right)$$




TT-ranks and timings (Oseledets-Khoromskij)

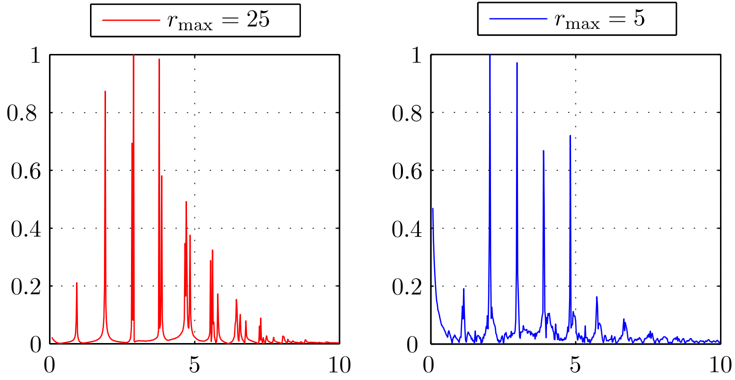
# SPECTRUM IN THE WHOLE

Use the evolution in time:

$$\frac{\partial \Psi}{\partial t} = iH\Psi, \quad \Psi(0) = \Psi_0.$$

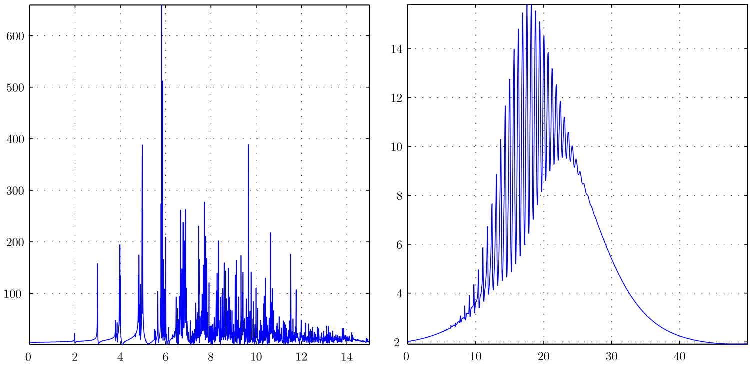
*Physical scheme* reads $\Psi(t) = e^{iHt}\Psi_0$, then we find the autocorrelation function $a(t) = (\Psi(t), \Psi_0)$ and its Fourier transform.

# SPECTRUM IN THE WHOLE




Henon-Heilse spectra for $f = 2$ and different TT-ranks.

# SPECTRUM IN THE WHOLE




Henon-Heiles spectra for $f = 4$ and $f = 10$.

# TT FOR EQUATIONS WITH PARAMETERS

Diffusion equation on $[0,1]^2$. The diffusion coefficients are constant in each of $p \times p$ square subdomains, i.e. $p^2$ parameters varing from 0.1 to 1.

256 points in each of parameters, space grid of size $256 \times 256$. The solution *for all values of parameters* is approximated by TT with relative accuracy $10^{-5}$:

| Number of parameters | Storage |
|---|---|
| 4 | 8 Mb |
| 16 | 24 Mb |
| 64 | 78 Mb |

## WTT FOR DATA COMPRESSION

$f(x) = \sin(100x)$

A signal on uniform grid with the stepsize $1/2^d$ on $0 \leqslant x \leqslant 1$ converts into a tensor of size $2 \times 2 \times \ldots \times 2$ with all TT-ranks $= 2$.

The Dobechis transform gives *much more* nonzeros:

| $\varepsilon$ | storage(WTT) | storage for filters | storage(D4) | storage(D8) |
|---|---|---|---|---|
| $10^{-4}$ | 2 | 152 | 3338 | 880 |
| $10^{-6}$ | 2 | 152 | 19696 | 2010 |
| $10^{-8}$ | 2 | 152 | 117575 | 6570 |
| $10^{-10}$ | 2 | 152 | 845869 | 15703 |
| $10^{-12}$ | 2 | 152 | 1046647 | 49761 |

$\sin(100x)$, $n = 2^d$, $d = 20$

# WTT FOR COMPRESSION OF MATRICES

WTT for *vectorized matrices* applies after reshaping:

$$a(i_1 \dots i_d \,;\, j_1 \dots j_d) \;\longrightarrow\; \tilde{a}(i_1 j_1 \,;\, \dots \,;\, i_d j_d).$$

WTT compression with accuracy $\varepsilon = 10^{-8}$ for the Cauchy-Hilbert matrix

$$a_{ij} = 1/(i-j) \text{ for } i \neq j, \quad a_{ii} = 0.$$

| $n = 2^d$ | storage(WTT) | storage(D4) | storage(D8) | storage(D20) |
|---|---|---|---|---|
| $2^5$ | 388 | 992 | 992 | 992 |
| $2^6$ | 752 | 4032 | 3792 | 3348 |
| $2^7$ | 1220 | 15750 | 13246 | 8662 |
| $2^8$ | 1776 | 59470 | 41508 | 20970 |
| $2^9$ | 2260 | 213392 | 102078 | 45638 |
| $2^{10}$ | 2744 | 780590 | 215738 | 95754 |
| $2^{11}$ | 3156 | 1538944 | 306880 | 176130 |

# TT IN DISCRETE OPTIMIZATION

Among all elements of a tensor given by TT find minimum or maximum. Discrete optimization problem is solved a an eigenvalue problem for diagonal matrices. Block minimization of Raleigh quotient in TT format, blocks of size 5, TT-ranks $\leqslant 5$ (O.S.Lebedeva).

| Function | Domain | Size | Iter. | $(Ax, x)$ | $(Ae_i, e_i)$ $e_i \approx x$ | Exact max |
|---|---|---|---|---|---|---|
| $\prod_{i=1}^{3}(1+0.1\,x_i+\sin x_i)$ | $[1, 50]^3$ | $2^{15}$ | 30 | 428.2342 | 429.2342 | 429.2342 |
| same | $[1, 50]^3$ | $2^{30}$ | 50 | 430.7838 | 430.7845 | |
| $\prod_{i=1}^{3}(x+\sin x_i)$ | $[1, 20]^3$ | $2^{15}$ | 30 | 8181.2 | 8181.2 | 8181.2 |
| same | $[1, 20]^3$ | $2^{30}$ | 50 | 8181.2 | 8181.2 | |

# CONCLUSIONS AND PERSPECTIVES

- TT algorithms (`http://pub.inm.ras.ru`) are efficient new instruments for compression of vectors and matrices. Storage and complexity depend on matrix size *logarithmically*.
- Free access to a current version of TT-library: `http://spring.inm.ras.ru/osel`.
- There are some theorems with TT-rank estimates. Sharper and more general estimates are to be derived. Difficulty is in nonlinearity of TT decompositions.

# CONCLUSIONS AND PERSPECTIVES

- TT interpolation methods provide new efficient methods for tabulation of functions of many variables, also those that are hard to evaluate.
- There are examples of application of TT methods for fast and accurate computation of multidimensional integrals.
- TT methods are successfully applied to image and signal processing and may compete with other known methods.

# CONCLUSIONS AND PERSPECTIVES

- TT methods are a good base for numerical solution of multidimensional problems of quantum chemistry, quantum molecular dynamics, optimization in parameters, model reduction, multiparametric and stochastic differential equations.