Tensor trains, TT and QTT formats

I. V. Oseledets

INM RAS, www.inm.ras.ru,http://spring.inm.ras.ru/osel, GERRUS LAB (Moscow-Leipzig),www.inm.ras.ru/gerrus.html

September 16, 2010

Tensor is a multidimensional array

$$\mathbf{A} = A(i_1, \ldots, i_d).$$

(= polylinear form).

Tamara Kolda, Brett Bader, Tensor decompositions and applications, SIAM Review (2009).

Do you do tensors?

History and motivation

Many years Tyrtyshnikov group studied approximation of large dense matrices

History and motivation

The most important matrix decomposition — Singular value decomposition, SVD

$$A \approx U \Lambda V^\top,$$

$$a_{ij} \approx \sum_{\alpha=1}^{r} u_{i\alpha} \lambda_\alpha v_{j\alpha}$$

History and motivation

Can it be generalized to 3-and d-dimensional case (orthogonality, computational algorithms...) ?

History and motivation

Separation of variables:

Canonical format:

$$A(i_1, i_2, i_3) \approx \sum_{\alpha=1}^{r} U_1(i_1, \alpha) U_2(i_2, \alpha) U_3(i_3, \alpha).$$

Approximation problem is ill-posed, no algorithms!

History and motivation

Another try: Tucker decomposition

$$A_(i_1, i_2, i_3) \approx$$
$$\sum_{\alpha_1,\alpha_2,\alpha_3} G(\alpha_1, \alpha_2, \alpha_3) U_1(i_1, \alpha_1) U_2(i_2, \alpha_2) U_3(i_3, \alpha_3).$$

History and motivation

Decomposition properties:

	Canonical	Tucker
Algorithms	No robust $(-)$	SVD-based $(+)$
Parameters	dnr $(+)$	$dnr + r^d$ $(-)$

History

Initial attempt used hierarchical separation of indices (Tyrtyshnikov & Oseledets, SISC, 2009)
Considered in a different view by Hackbusch, Kuhn, and Grasedyck (H-Tucker format)

History

Initial attempt used hierarchical separation of indices (Tyrtyshnikov & Oseledets, SISC, 2009)
Considered in a different view by Hackbusch, Kuhn, and Grasedyck (H-Tucker format)
Oseledets, Tyrtyshikov, LAA, 2010 Tensor tree decomposition does not need a tree

History

Two formats are algebraically equivalent

History

Two formats are algebraically equivalent

Bet

I will pay 10 Euro

to a person that sends me a tensor that:

1. Fits into the memory of my laptop
2. Has small ranks for some H-Tucker
3. In two days I will not send a permutation of dimensions that has the same TT-ranks

TT-format

Oseledets, 2009:

TT-format

$$A(i_1, i_2, \ldots, i_d) \approx$$
$$\sum_{\alpha_1, \alpha_2, \ldots, \alpha_{d-1}} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \ldots G_d(\alpha_{d-1}, i_d)$$

TT-format

Oseledets, 2009:

TT-format

$$A(i_1, i_2, \ldots, i_d) \approx$$
$$\sum_{\alpha_1, \alpha_2, \ldots, \alpha_{d-1}} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \ldots G_d(\alpha_{d-1}, i_d)$$

TT-format: discrete arrays

$$A(i_1, \ldots, i_d) = G_1(i_1) G_2(i_2) \ldots G_d(i_d)$$

TT-format

TT-format: discrete arrays

$$A(i_1, \ldots, i_d) = G_1(i_1) G_2(i_2) \ldots G_d(i_d)$$

$G_k(i_k)$ is a $r_{k-1} \times r_k$ matrix, $r_1 = r_d = 1$.

TT-format

TT-format: discrete arrays

$$A(i_1, \ldots, i_d) = G_1(i_1) G_2(i_2) \ldots G_d(i_d)$$

Oseledets, 2010:

FTT-format: separation of variables

$$f(x_1, \ldots, x_d) = g_1(x_1) g_2(x_2) \ldots g_d(x_d)$$

TT-format

$$f = x_1 + x_2 + \ldots + x_d$$

Canonical rank: d

TT-rank: 2

$$f = \begin{pmatrix} x_1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ x_2 & 1 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} 1 & 0 \\ x_{d-1} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ x_d \end{pmatrix}$$

TT in a nutshell

There are five basic elements that make TT effective

TT in a nutshell

1. TT-ranks are matrix ranks, TT-SVD algorithm
2. All basic arithmetic, linear in d, polynomial in r
3. Fast TENSOR ROUNDING
4. TT-cross method, exact interpolation formula
5. Quantics-TT decomposition — binarization (or tensorization) of vectors, matrices

TT in a nutshell

First element : TT-rank is a matrix rank

Th: Exists decomposition with $r_k = \mathbf{rank}\, A_k$

A_k is the k-th unfolding matrix:

$$A_k = A(i_1 i_2 \dots i_k; i_{k+1} \dots i_d)$$

in MATLAB:

$$A_k = \mathbf{reshape}(\mathbf{A}, [\prod_{s=1}^{k} n_s, \prod_{s=k+1}^{d} n_s])$$

TT in a nutshell

Th: Exists decomposition with $r_k = \mathbf{rank}\, A_k$
Proof is constructive and is given by
TT-SVD algorithm (Oseledets, 2009)

Require: d-dimensional tensor $\mathbf{A}$ required accuracy ε
Ensure: Cores $G_1, \ldots, G_d$ of the TT-approximation $\mathbf{B}$ to $\mathbf{A}$ in the TT-format with smallest possibles compression ranks $\widehat{r}_k$ such that

$$||\mathbf{A} - \mathbf{B}||_F \leq \varepsilon ||\mathbf{A}||_F,$$

{Initialization}
Compute truncation parameter $\delta = \frac{\varepsilon}{\sqrt{d-1}} ||\mathbf{A}||_F$.
Temporary tensor: $\mathbf{C} = \mathbf{A}$.
$N = \mathrm{numel}(A)$, $r_0 = 1$.
for $k = 1$ to $d - 1$ **do**
  $C := \mathrm{reshape}(C, [r_{k-1} n_k, \frac{N}{r_{k-1} n_k}])$.
  Compute δ-truncated SVD: $C = USV + E, ||E||_F \leq \delta, \quad r_k = \mathrm{rank}_\delta(C)$.
  New core: $G_k := \mathrm{reshape}(U, [r_{k-1}, n_k, r_k])$.
  $C := SV^\top$.
  $N := \frac{N r_k}{n_k r_{k-1}}$.
end for
$G_d = C$.
Return tensor $\mathbf{B}$ in TT-format with cores $G_1, \ldots, G_d$.

TT in a nutshell

Maybe someone noticed "magic" factor $\sqrt{d-1}$
In follows from stability result:
If ε_k is the truncation error at k-th step, then
TT-SVD computes approximation $\mathbf{B}$ which satisfy

$$\|\mathbf{A} - \mathbf{B}\|_F \leq \sqrt{\sum_{k=1}^{d-1} \varepsilon_k^2}.$$

Second element

Second element: Fast & trivial linear algebra:
Addition, Hadamard product, scalar product, convolution
All scale linear in d

Third element

Third element: Tensor rounding:

Given a tensor $\mathbf{A}$ in TT-format, approximate in with $\mathbf{B}$ with smaller ranks, maintaining accuracy ε

$$\|\mathbf{A} - \mathbf{B}\|_F \leq \varepsilon \|\mathbf{B}\|_F$$

It can be done in $\mathcal{O}(dnr^3)$ operations

Third element

Require: d-dimensional tensor $\mathbf{A}$ in the TT-format, required accuracy ε
Ensure: $\mathbf{B}$ in the TT-format with smallest possibles compression ranks $\widehat{r}_k$ such that

$$||\mathbf{A} - \mathbf{B}||_F \leq \varepsilon ||\mathbf{A}||_F, \quad i.e. \quad \mathbf{B} = T_\varepsilon(\mathbf{A}).$$

Let G_k, $k = 1, \ldots, d-1$ be cores of $\mathbf{A}$.
{Initialization}
Compute truncation parameter $\delta = \frac{\varepsilon}{\sqrt{d-1}} ||\mathbf{A}||_F$.
{Right-to-left orthogonalization}
for $k = d$ to 2 step -1 **do**
    $[G_k(\beta_{k-1}; i_k \beta_k), R(\alpha_{k-1}, \beta_{k-1})] := \mathrm{QR}_{\mathrm{rows}}(G_k(\alpha_{k-1}; i_k \beta_k))$.
    $G_{k-1} := G_k \times_3 R$.
end for
{Compression of the orthogonalized representation}
for $k = 1$ to $d-1$ **do**
    {Compute δ-truncated SVD}
    $[G_k(\beta_{k-1} i_k; \gamma_k), \Lambda, V(\beta_k, \gamma_k)] := \mathrm{SVD}_\delta[G_k(\beta_{k-1} i_k; \beta_k)]$.
    $G_{k+1} := G_{k+1} \times_1 (V\Lambda)^\top$.
end for
Return G_k, $k = 1, \ldots, d$ as cores of $\mathbf{B}$.

Third element

TENSOR ROUNDING : Reduce ranks in $\mathcal{O}(dnr^3)$ operations while

$$\|\mathbf{A} - \mathbf{B}\|_F \leq \varepsilon \|\mathbf{B}\|_F$$

Fourth element

Fourth element: Cross approximation method

The problem at the basic level is solved!

Fourth element

Fourth element: Cross approximation method
The problem at the basic level is solved!

Oseledets-Tyrtyshnikov, 2009:
You can exactly interpolate rank-r tensor on $\mathcal{O}(dnr^2)$ elements

Fourth element

Oseledets-Tyrtyshnikov, 2009:
You can exactly interpolate rank-r tensor on $\mathcal{O}(dnr^2)$ elements

TT interpolation formula

$$A(i_1, i_2, \ldots, i_d) = \sum_{\alpha_1, \ldots, \alpha_{d-1}} \widehat{C}_1(i_1, \alpha_1) \widehat{C}_2(\alpha_1, i_2, \alpha_2) \ldots \widehat{C}_d(\alpha_{d-1}, i_d),$$

where $\widehat{C}_k = C_k \times_3 \widehat{A}_k^{-1}$

$$C_k(\alpha_{k-1}, i_k, \alpha_k) = A(\mathcal{I}_k(\alpha_{k-1}), i_k, \mathcal{J}(\alpha_k))$$

$$\widehat{A}_k(\alpha_k, \beta_k) = A(\mathcal{I}_k(\alpha_k), \mathcal{J}_k(\beta_k))$$

$$\mathcal{I}_k(\alpha) = [i_1^{(\alpha,k)}, \ldots, i_k^{(\alpha,k)}]$$

$$\mathcal{J}_k(\alpha) = [i_{k+1}^{(\alpha,k)}, \ldots, i_d^{(\alpha,k)}]$$

$\widehat{A}_k$ should be nonsingular

$\mathcal{I}_k$ should be left-nested

Fourth element

Just one example:

Example (G. Bonik)

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

«Dump» idea: approximate by rectangle rule.

Required number of nodes: 2^{77}.

Ranks of QTT-decomposition — 12, time — 1 sec.

Fifth element

Fifth element: QTT-decomposition

Fifth element

Fifth element: QTT-decomposition

From previous example:

$v_k = f(x_k), k = 1, \ldots, 2^d, d = 20, 30, 40, 50, 60, 70...$

Fifth element

Fifth element: QTT-decomposition

From previous example:

$$v_k = f(x_k), k = 1, \ldots, 2^d, d = 20, 30, 40, 50, 60, 70...$$

Reshape into $2 \times 2 \times \ldots \times 2$ tensor:

$$\mathbf{A} = \mathbf{reshape}(v, 2 * \mathbf{ones}(1, d))$$

And compute its TT-decomposition

Fifth element

From previous example:

$v_k = f(x_k), k = 1, \ldots, 2^d, d = 20, 30, 40, 50, 60, 70...$

Reshape into $2 \times 2 \times \ldots \times 2$ tensor:

$\mathbf{A} = \mathbf{reshape}(v, 2 * \mathbf{ones}(1, d))$

And compute its TT-decomposition

Cores are $r_{k-1} \times 2 \times r_k$

Explicit representations

Known: $f(x) = e^{\lambda x}$

Rank estimates: $f(x)$ is a polynomial, or piece-wise polynomials

In a new preprint 2010-04 (Oseledets) explicit formula for TT & QTT decompositions are obtained:

Explicit representations

Introduce new separation of variables:

Functional TT decomposition

$$f(x_1, \ldots, x_d) = \sum_{\alpha_1, \ldots, \alpha_{d-1}} g_1(x_1, \alpha_1) g_2(\alpha_1, x_2, \alpha_2) \ldots g_d(\alpha_{d-1}, x_d)$$

Functional TT decomposition-2

$$f = g_1(x_1) G_2(x_2) \ldots G_{d-1}(x_{d-1}) g_d(x_d)$$

Looks like rank-1, but G_k are matrices, g_1, g_d are vectors!

Explicit representations

Functional TT decomposition-2

$$f = g_1(x_1)G_2(x_2)\dots G_{d-1}(x_{d-1})g_d(x_d)$$

Example 1:

For a function

$$f(x_1,\dots,x_d) = w_1(x_1) + w_2(x_2) + \dots + w_d(x_d),$$

FTT-decomposition has form

$$f = \begin{pmatrix} w_1(x_1) & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ w_2(x_2) & 1 \end{pmatrix} \cdot \dots \cdot \begin{pmatrix} 1 & 0 \\ w_{d-1}(x_{d-1}) & 1 \end{pmatrix} \begin{pmatrix} 1 \\ w_d(x_d) \end{pmatrix}$$

Explicit representations

Functional TT decomposition-2

$$f = g_1(x_1)\, G_2(x_2) \ldots G_{d-1}(x_{d-1}) g_d(x_d)$$

Example 2:

For a function

$$f(x_1, \ldots, x_d) = \sin(x_1 + x_2 + \ldots + x_d)$$

FTT-decomposition has form

$$f = \begin{pmatrix} \sin x_1 & \cos x_1 \end{pmatrix} \begin{pmatrix} \cos x_2 & -\sin x_2 \\ \sin x_2 & \cos x_2 \end{pmatrix} \cdot \ldots \cdot$$

$$\begin{pmatrix} \cos x_{d-1} & -\sin x_{d-1} \\ \sin x_{d-1} & \cos x_{d-1} \end{pmatrix} \begin{pmatrix} \cos x_d \\ \sin x_d \end{pmatrix}.$$

Explicit representations

Functional TT decomposition-2

$$f = g_1(x_1) G_2(x_2) \dots G_{d-1}(x_{d-1}) g_d(x_d)$$

Relation to QTT:

$$v_k = f(p_k) = f(a + h(i_0 + 2i_1 + 4i_2 + \dots + 2^{d-1} i_d)) =$$
$$f(x_0 + \dots + x_{d-1}$$
$$x_k = \frac{a}{d} + 2^{k-1} i_k h$$

Explicit representations

Obtained explicit formulae for:
polynomials, sine function, Hankel functions (i.e. $f(x+y)$ has low separation rank)

Relation to wavelets

Can be interpreted as subspace approach!

Relation to wavelets

Can be interpreted as subspace approach!

$$v = f(x_k)$$

$$A = \mathbf{reshape}(v, [2, n/2])$$

$A = USV$ — truncated SVD, and then compress r rows of

$$B = SV = U^\top A$$

Relation to wavelets

What is $U^\top A$? It is application of a local 2×2 filter to each subvector!

For example, if $U = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

It is a Haar transform!

Relation to wavelets

Wavelets are particular case of the general scheme

How to determine U, ranks r_k, is SVD optimal?

Matrices U_k are *adaptive linear filters*

Relation to wavelets

Consider a sine function:

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

Sine function

ε	nnz(WTT)	Memory(U_k)	nnz(D4)	nnz(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

Relation to wavelets

Any function f gives a wavelet tensor train (WTT) transform

Theorem: It is always orthogonal

Experiment: $f(x) = x^k$ gives $(k+1)$ zero moments!

Matrix compression, Hilbert matrix

ε	nnz(WTT(x))	nnz(WTT(x^3))	nnz(D4)	nnz(D8)
10^{-4}	43732	46546	54002	35456
10^{-6}	95898	71356	194188	66740
10^{-8}	196518	116238	588914	141794
10^{-10}	376378	182716	814022	291732
10^{-12}	623212	283972	926328	549962

Relation to wavelets

Generalized to two dimensions:
$A(i,j) \longrightarrow A(i_1, i_2, \ldots, i_d, j_1, \ldots, j_d) \longrightarrow$
$A(i_1, j_1, \ldots, i_d, j_d)$ And do the same (multiply by $U^\top$, compress rows of result)
Non-separable 2D transform!

Relation to wavelets

Hilbert matrix again

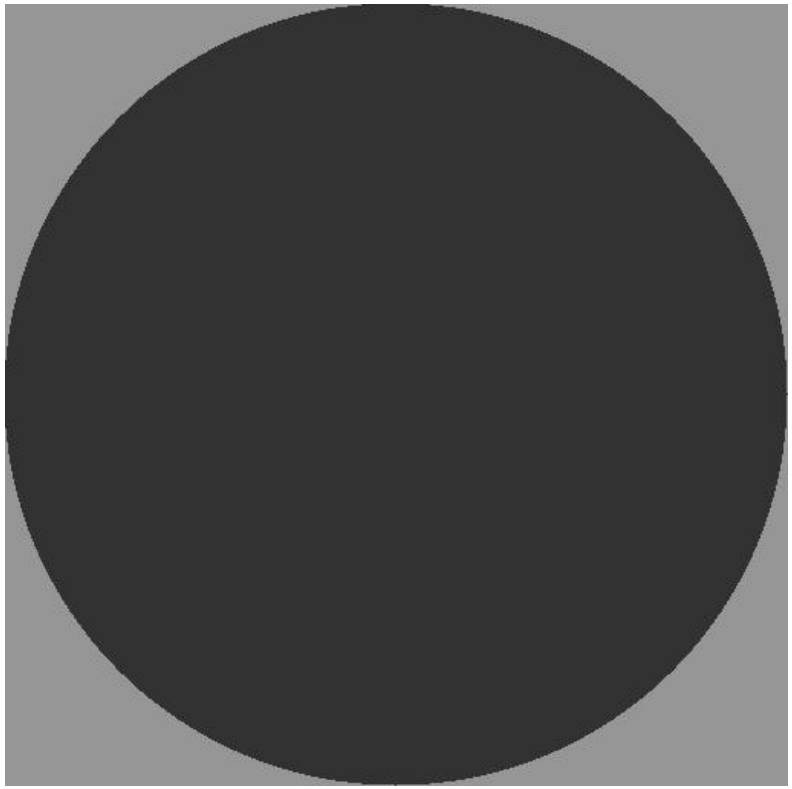
Hilbert-2D WTT

$n = 2^d$	Memory(WTT)	nnz(D4)	nnz(D8)	nnz(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

Relation to wavelets

Cases where ε-strategy does not work:

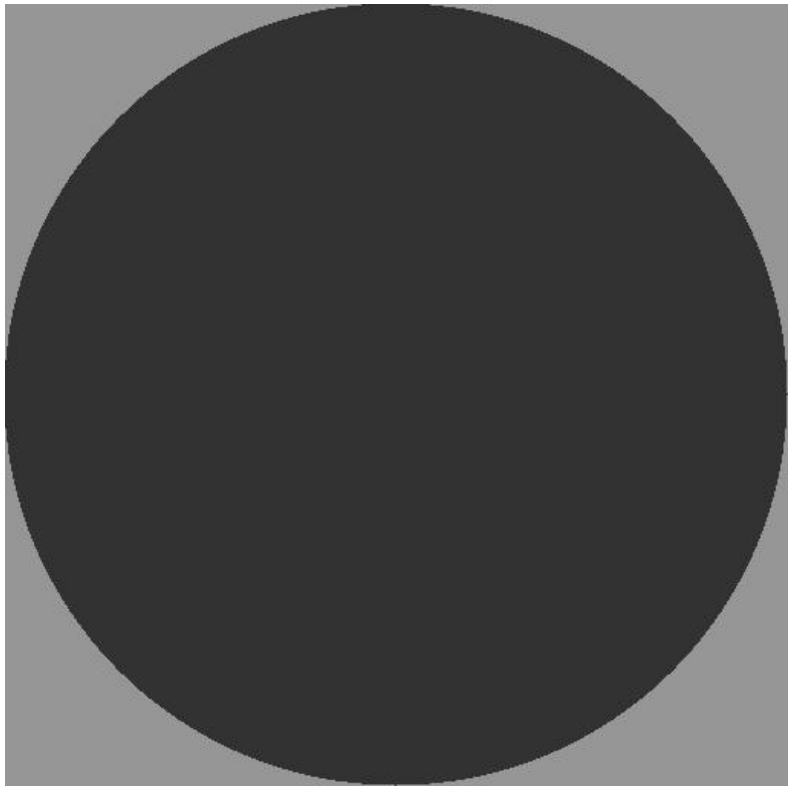
Disk



70 KB (WTT-ε) vs 8 KB (JPEG)

Relation to wavelets

Mixed does work: (leave 1-singular vector, other are sparse!)



50 bytes (WTT-`rmax`) vs 8 KB (JPEG)

Molecular dynamics and DMRG

Joint work with Boris Khoromskij

High-dimensional problems arise in quantum chemisty.

Molecular Schroedinger equation:

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

Describes vibrational motion of molecules.

V — Potential energy surface

Molecular dynamics and DMRG

2 problems: approximation (representation) V

Find eigenfunctions

Molecular dynamics and DMRG

Empiric approximations: Lennard-jones potential:
Atoms interact pairwise: $V_{ij} = A\left(-(\frac{B}{R})^6 + (\frac{C}{R})^{12})\right)$

Molecular dynamics and DMRG

Another approach: Vibrations around stable configuration:

$$V(R_1, \ldots, R_f) = V_* + \sum_{ij} \frac{\partial^2 V}{\partial R_i \partial R_j} \Delta R_i \Delta R_j + \ldots$$

Molecular dynamics and DMRG

Another approach: Vibrations around stable configuration:

$$V(R_1, \ldots, R_f) = V_* + \sum_{ij} \frac{\partial^2 V}{\partial R_i \partial R_j} \Delta R_i \Delta R_j + \ldots$$

Second-order approximations — harmonic approximation & analytical solution

Higher order gives correction

Resume: Polynomial potentials are widely used! (give good approximations)

The biggest molecule — protein BPT1 (Nature, 1996) contains thousands of molecules, 4-rth order polynomial.

Molecular dynamics and DMRG

And it looks like:
(Video borrowed for the site of Joel Bowman)
Vibrational motion of $H_5O_2^+$

Molecule

Known approaches

Approximation for V — it can be piecewise-polynomial
Eigenfunction has (n^N) parameters.

Known approaches

Chemists:

1. Canonical — rank-1 + perturbation theory (VSCF)
2. Multiconfigurational time-dependent Hartree method: (Meyer) — Tucker format (dimension 15).

Our approach

Introduce 2^d-grid and get df-dimensional problem.

Our approach

Introduce 2^d-grid and get df-dimensional problem.
Form H in QTT format

$$H(i_1,\ldots,i_d,j_1,\ldots,j_d) = \sum_{\alpha_1,\ldots,\alpha_{d-1}} H_1(i_1 j_1,\alpha_1) H_2(\alpha_1, i_2 j_2, \alpha_2)\ldots,$$

Our approach

Form H in QTT format

$$H(i_1,\ldots,i_d,j_1,\ldots,j_d) = \sum_{\alpha_1,\ldots,\alpha_{d-1}} H_1(i_1 j_1,\alpha_1) H_2(\alpha_1, i_2 j_2, \alpha_2)\ldots,$$

How to find eigenfunction?

Our approach

How to find eigenfunction? Physicists come to help:

Density Matrix Renormalization Group

Proposed for quantum spin systems 1992 - DMRG,

Steven White. 1300 papers.

Relaxation algorithm in QTT format! (R. Schneider)

Our approach

Physicists come to help:

Density Matrix Renormalization Group

Proposed for quantum spin systems 1992 - DMRG, Steven White. 1300 papers.

Relaxation algorithm in QTT format! (R. Schneider)

Idea: in QTT-decomposition fix everything except for two cores

G_k and G_{k+1}.

Our approach

$$(H\Psi, \Psi) \to \min, \text{ по } G_k(\alpha_{k-1}, i_k, \alpha_k),$$
$$G_{k+1}(\alpha_k, i_{k+1}, \alpha_{k+1})$$

Introduce supercore (superblock):

$$W = W(\alpha_{k-1}, i_k, i_{k+1}, \alpha_{k+1})$$

In W — eigenvalue problem with $4r^2$ unknowns

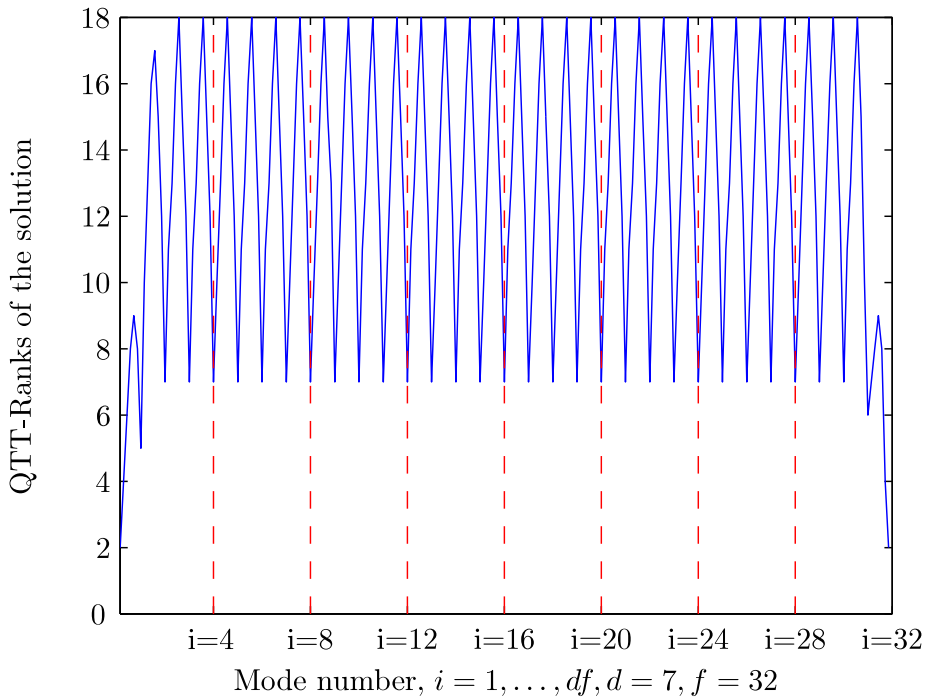
Use LOBCG (Knyazev) to solve eigenvalue problem + «smart initial condition»

Convergence in $3 - 5 - 7$ sweeps!

Henon-Heiles potential

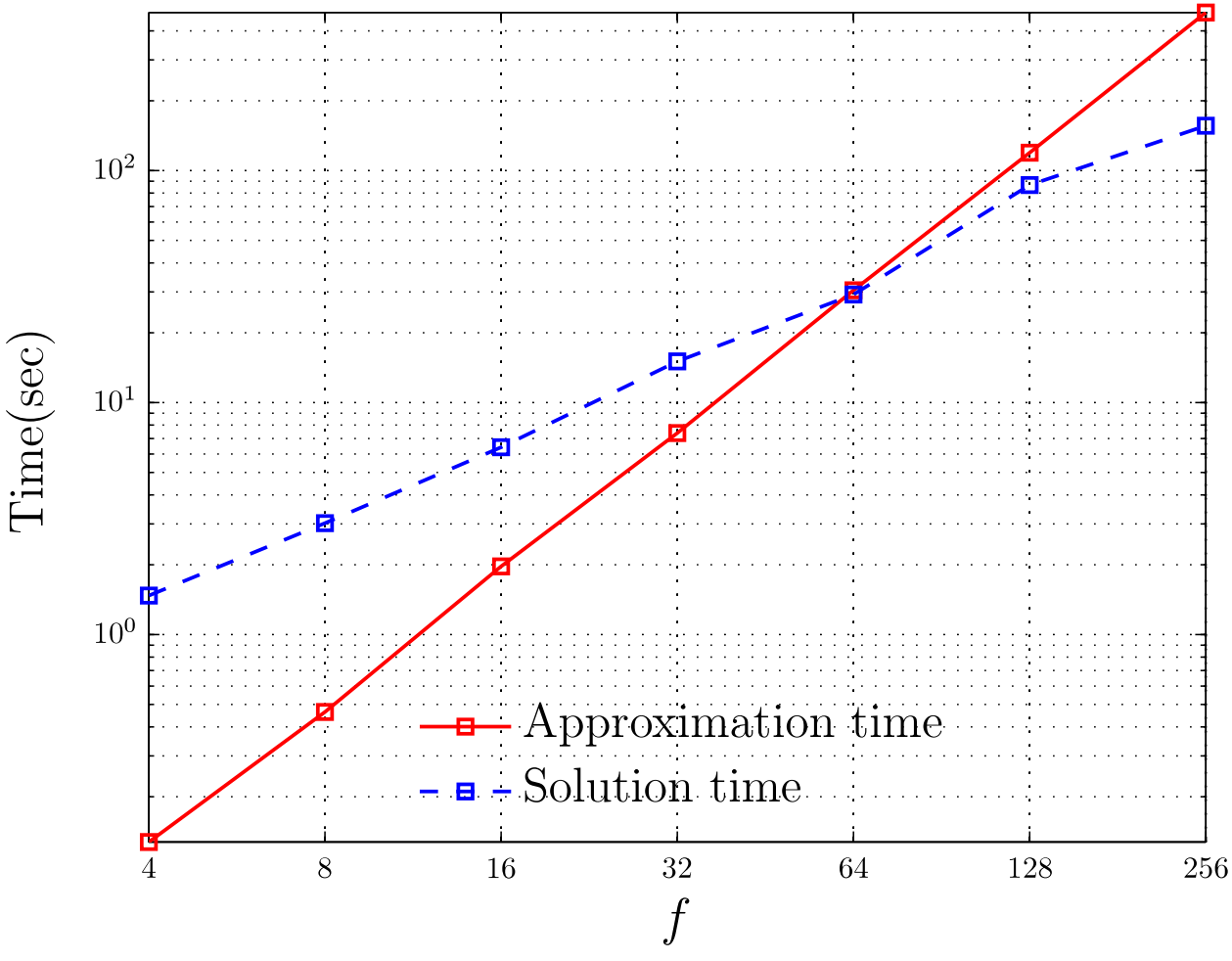
Henon-Heiles potential (classical test case):

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



QTT-ranks of solution

Henon-Heiles potential



Solution time

Bowman example

$$H = H_x + H_y + H_z + V_c(x, y, z),$$

where

$$\begin{aligned} h_x &= \frac{1}{2}\left(-\frac{\partial^2}{\partial x^2} + w_x^2 x^2 + 2\lambda\eta x^3\right), \\ h_y &= \frac{1}{2}\left(-\frac{\partial^2}{\partial x^2} + w_y^2 y^2 + 2\mu\zeta y^3\right), \\ h_z &= \frac{1}{2}\left(-\frac{\partial^2}{\partial x^2} + w_z^2 z^2\right). \end{aligned}$$

,

and

Bowman example

First eigenvalue computation

3-mode problem

n	$\lambda(10^{-2}) - \lambda^*$	$\lambda(10^{-4}) - \lambda^*$	$\lambda(10^{-7}) - \lambda^*$
2^4	-0.0688173	-0.0689082	-0.0689082
2^5	-0.0163878	-0.0176922	-0.0176922
2^6	-0.0030445	-0.0045148	-0.0045155
2^7	0.0004258	-0.0005373	-0.0011436
2^8	0.0025507	0.0003197	-0.0002879
2^9	0.0238319	0.0005368	-0.0000715
2^{10}	0.0244351	0.0005914	-0.0000173

Meyer example

H.-D. Meyer provided us with realistic approximation to the HONO potential (6-mode problem):

$$V(R_1, R_2, R_3, \theta_1, \theta_2, \tau) = S_0 + e^{-D(R_1, R_2, R_3, \theta_1, \theta_2)} \times \sum_{ijklmn} c_{ijklmn} Q_1^i Q_2^j Q_3^k Q_4^l Q_5^m \cos(nQ_6),$$

with

$$D = \sum_{i=1}^{3} d_i (R_i - R^{\mathrm{ref}})_i^2 + \sum_{j=1}^{2} d_{j+3} (\theta_j - \theta^{\mathrm{ref}_j})^2,$$

and

$$Q_{1/2/3} = 1 - e^{-0.7(R_{1/2/3} - R^{\mathrm{ref}}_{1/2/3})}, Q_{4/5} = \theta_{1/2} - \theta^{\mathrm{ref}}_{1/2}, Q_6 = \tau - \tau^{\mathrm{ref}}.$$

Meyer example

Approximation ranks

n	$\mathbf{rank}(10^{-2})$	$\mathbf{rank}(10^{-4})$	$\mathbf{rank}(10^{-7})$
2^4	3.2	7.3	12.0
2^5	3.3	7.9	12.5
2^6	3.4	8.0	12.6
2^7	3.4	8.1	12.7
2^8	3.4	8.1	12.7
2^9	3.7	8.0	12.7
2^{10}	3.6	8.0	12.7

Conclusion

- TT-format — stable algorithms, no curse of dimensionality
- Quantics TT format = algebraic wavelet transform
- New communities: molecular dynamics, DMRG

References

1. Oseledets I.V., Tyrtyshnikov E.E., Breaking the curse of dimensionality, or how to use SVD in many dimensions, SISC, 2009
2. Oseledets I.V., Approximation of $2^d \times 2^d$ matrices using tensor decomposition, SIMAX, 2010
3. Oseledets I.V., Compact matrix form of d-dimensional tensor decomposition, SISC, 2010 (submitted)
4. Oseledets I.V., Khoromskij B.N., QTT: Bridging the gap between molecular dynamics and DMRG (in preparation)
5. Oseledets I.V., Tyrtyshnikov E.E., Algebraic wavelet transform via quantics tensor train decomposition, INM RAS Preprint, 2010-03
6. Oseledets I.V., Constructive representation of functions in tensor formats, INM RAS Preprint, 2010-04

References

Papers and MATLAB code (TT Toolbox 1.0) can be downloaded from

http://spring.inm.ras.ru/osel

TT Toolbox 1.1 is available by request