Numerical tensor methods and their applications

I.V. Oseledets

14 May 2013

All lectures

4 lectures,

- 2 May, 08:00 - 10:00: Introduction: ideas, matrix results, history.
- 7 May, 08:00 - 10:00: Novel tensor formats (TT, HT, QTT).
- 8 May, 08:00 - 10:00: Advanced tensor methods (eigenproblems, linear systems).
- 14 May, 08:00 - 10:00: Advanced topics, recent results and open problems.

Brief recap of Lecture 3

- QTT-format: explicit representations of functions
- QTT-format: explicit representation of operators
- QTT theorem
- QTT-Fourier transform as a tensor network
- QTT-convolution
- Linear systems
- Eigenvalue problems

Plan of lecture 4

- Dynamical low-rank approximation
- Solving non-stationary problems
- Solving multiparametric problems
- Applications to multiparametric problems

Non-stationary problems

Basic setting:

$$\frac{dy}{dt} = F(y,t), \quad y(0) = y_0$$

$$y \to Y(i_1, \ldots, i_d)$$

Application: spectra

Of the main application is the computation of
vibrational spectra

Application: spectra

Problem setting:

$$\frac{d\psi}{dt} = iH\psi, \quad \psi(0) = \psi_0.$$

- $\psi = \psi(q_1, \ldots, q_f)$ — wavefunction
- $H = -\frac{1}{2}\Delta + V(q_1, \ldots, q_f)$.
- $\Delta = \sum_{i=1}^{f} \frac{\partial^2}{\partial q_i^2}$.
- V — potential energy surface
- $q_1, \ldots, q_f$ — degrees of freedom

Application: spectra

Initial setting is the Scrödinger equation

1. Fix the positions of the nuclei — get the energy $V(R_1, \ldots, R_N)$
2. Find the minima, expand V again the minima into Taylor series.
3. (optional): Find normal coordinates, $f = 3N - 6$ (N is the number of atoms)

Main equation

$$\frac{d\psi}{dt} = i(-\frac{1}{2}\Delta + V)\psi, \quad \psi(0) = \psi_0.$$

Discretize on a grid. Laplace has TT-ranks equal to 2 and QTT-ranks bounded by 4.

Main equation

What about V?

Theorem (Rank estimate)

For the polynomial of form

$$V(q_1, \ldots, q_f) = \sum_{i_1, \ldots, i_s = 1}^{f} a(i_1, \ldots, i_s) \prod_{k=1}^{s} q_{i_k},$$

$$\mathbf{rank}_{TT}(V) = C_0 f^{[\frac{s}{2}]} + o(f^{[\frac{s}{2}]}).$$

Main equation

$$\frac{d\psi}{dt} = i(-\frac{1}{2}\Delta + V)\psi, \quad \psi(0) = \psi_0.$$

Discretize on a grid. Laplace has TT-ranks equal to 2 and QTT-ranks bounded by 4.
It is easy to put $H = -\frac{1}{2}\Delta + V$ into the TT-format.

But how it is connected to the computation of spectra?

Eigenvalues via spectra

We want to compute:

$$H\psi = \lambda\psi,$$

But what we do:

1. Take some ψ_0
2. Solve $\frac{dpsi}{dt} = H\psi, \quad \psi(0) = \psi_0.$
3. Compute \$a(t) = (\psi(t), \psi(0))
4. Find FFT of $a(t)$.

Eigenvalues via spectra

$$\psi(t) = \exp(iHt)\psi_0$$

$$\psi_0 = \sum_{k=0}^{\infty} c_k \phi_k,$$

$$H\phi_k = \lambda_k \phi_k,$$

$$a(t) = \sum_{k=0}^{\infty} |c_k|^2 \exp(i\lambda_k t)$$

Thus, the FFT of $a(t)$ gives pikes at λ_k (try that!)

Why should you do that?

Eigenvalues via spectra

How to select initial condition?

$$\psi_0 = \sum_{k=0}^{\infty} c_k \phi_k,$$

c_k should not decay fast!

Example: harmonic oscillator,

Typically $\psi_0 = \prod_{k=1}^{f} \exp(-\alpha_k (q_k^{(0y)} - q_k)^2)$.

Problems and solutions

$$\frac{d\psi}{dt} = i(-\frac{1}{2}\Delta + V)\psi, \quad \psi(0) = \psi_0.$$

It is a hyperbolic problem ($\approx$ square root of the wave equation)

The ranks grow with $t \longrightarrow \infty$ (can become chaotic, boundary reflections, etc.)

No chance?

Dynamical approximation

The idea of the dynamical approximation: replace $\psi(t)$ by its low-parametric representation $\widehat{\psi}(t)$ and hope that $a(t)$ has the same statistics

How to find $\psi(t)$?

Dynamical approximation

The idea of the dynamical approximation: replace $\psi(t)$ by its low-parametric representation $\widehat{\psi}(t)$ and hope that $a(t)$ has the same statistics

How to find $\psi(t)$?

We need an optimization principle

The answer: Dirac-Frenkel variational principle!

Manifold

S Holtz, T Rohwedder, R Schneider - Numerische Mathematik, On manifolds of tensors of fixed TT-rank

The set of tensors with bounded TT-ranks forms a manifold

$$\mathrm{TT}_{\mathbf{r}}, \quad \mathbf{r} = (r_1, \ldots, r_{d-1})$$

Tangent space

We can define a tangent space $\mathcal{M}(\mathbf{TT_r})$ at a point X

Do it only in 2D (C. Lubich)

$$A = USV^\top$$

$$\delta A = \delta U S V^\top + U \delta S V^\top + U S \delta V^\top$$

Projector onto the tangent space:

$$P_X(\delta A) = \delta A - (I - UU^\top)\delta A (I - VV^\top)$$

Tangent space(2)

Why should we care about the tangent space?

Because of the variational principle

Dirac-Frenkel

Given $A(t)$, the dynamical (low-rank) approximation is defined as $X(t)$

$$(\frac{dA}{dt} - \frac{dX}{dt}, \delta X) = 0, \quad \delta X \in \mathcal{T}_X(\mathcal{M})$$

The velocity is in the tangent space!

Dynamical low-rank appr. of matrices

The equations for U, S, V:

$$\frac{dU}{dt}(t) = (I - U(t)U(t)^\top)\frac{dA}{dt}(t)V(t)S(t)^{-1}$$
$$\frac{dV}{dt}(t) = (I - V(t)V(t)^\top)\frac{dA}{dt}(t)^\top U(t)S(t)^{-\top}$$
$$\frac{dS}{dt}(t) = U(t)^\top \frac{dA}{dt}(t)V(t).$$

Dynamical low-rank appr. of matrices

$$\frac{dX}{dt} = P_X\left(\frac{dA}{dt}\right)$$

MCTDH

MCTDH (H.-D. Meyer)

MultiConfigurational Time-dependent Hartree Method

Uses dynamical low-rank approximation in the Tucker format!

Equations for the factors, equations for the core

How to integrate them well?

Problems with dynamical low-rank integration

The main problem of the dynamical equations is the inversion of S

But the equation for X does not have this problem!

$$P_X(Z) = Z - (I - UU^\top)Z(I - VV^\top)$$

KSL integrator

C. Lubich, O., Projector-splitting integrator for the dynamical low-rank approximation of matrices

Splitting

$$P(Y)Z = ZP_{\mathcal{R}(Y^\top)} + P_{\mathcal{R}(Y)}Z - P_{\mathcal{R}(Y)}Z$$

Algorithm:

- K-step: $\frac{d(US)}{dt} = \frac{dA}{dt}V$
- QR: $K_1 = U_1\widehat{S}_1$
- L-step: $\frac{d(VS^\top)}{dt} = \frac{dA}{dt}^\top U$
- QR: $L_1 = U_1\widetilde{S}_1$
- S-step: $\frac{dS}{dt} = -U^\top\frac{dA}{dt}V$ (backward in time!)

Splitting

$$P(Y)Z = ZP_{\mathcal{R}(Y^\top)} + P_{\mathcal{R}(Y)}Z - P_{\mathcal{R}(Y)}Z$$

Algorithm:

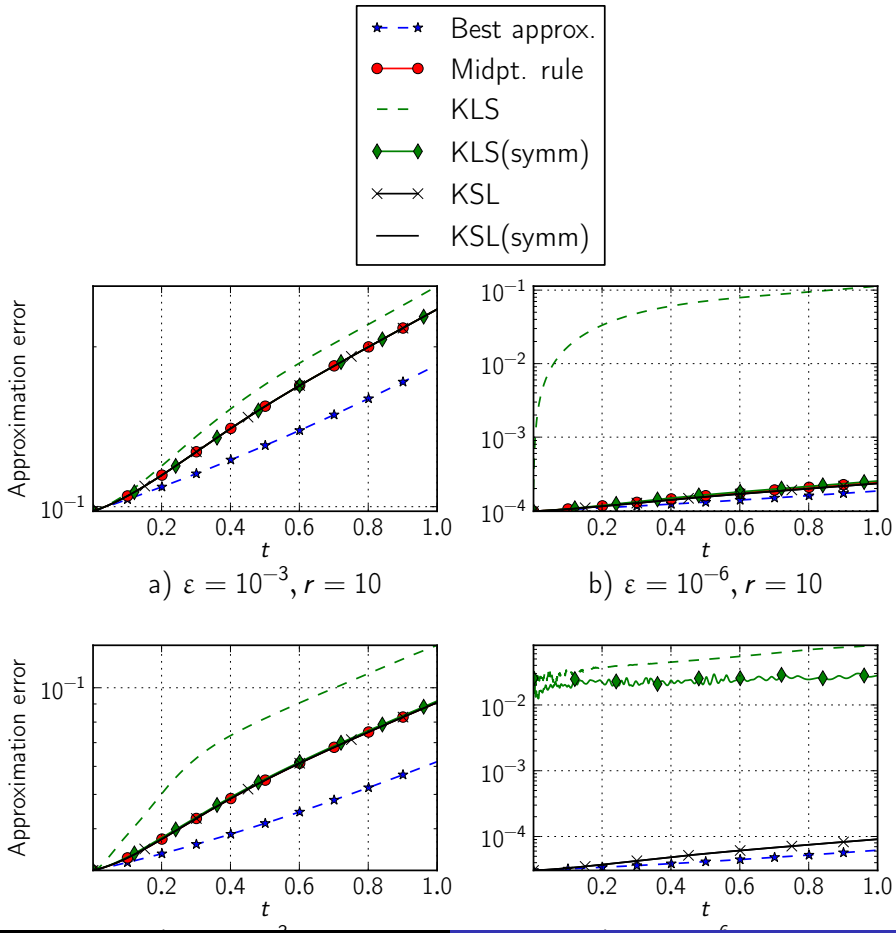
- K-step: $\frac{d(US)}{dt} = \frac{dA}{dt}V$
- QR: $K_1 = U_1\widehat{S}_1$
- L-step: $\frac{d(VS^\top)}{dt} = \frac{dA}{dt}^\top U$
- QR: $L_1 = U_1\widetilde{S}_1$
- S-step: $\frac{dS}{dt} = -U^\top \frac{dA}{dt}V$ (backward in time!)

First order explicit scheme, KLS

Splitting

The K-S-L ordering is much better

Experiments on rank-10 matrix A with noise ε.



Theory

Theorem (Exactness)

Suppose that $A(t)$ has rank at most r for all t. With the initial value $Y_0 = A(t_0)$, the splitting algorithm is then exact: $Y_1 = A(t_1)$.
From this fact we can prove overapproximation result

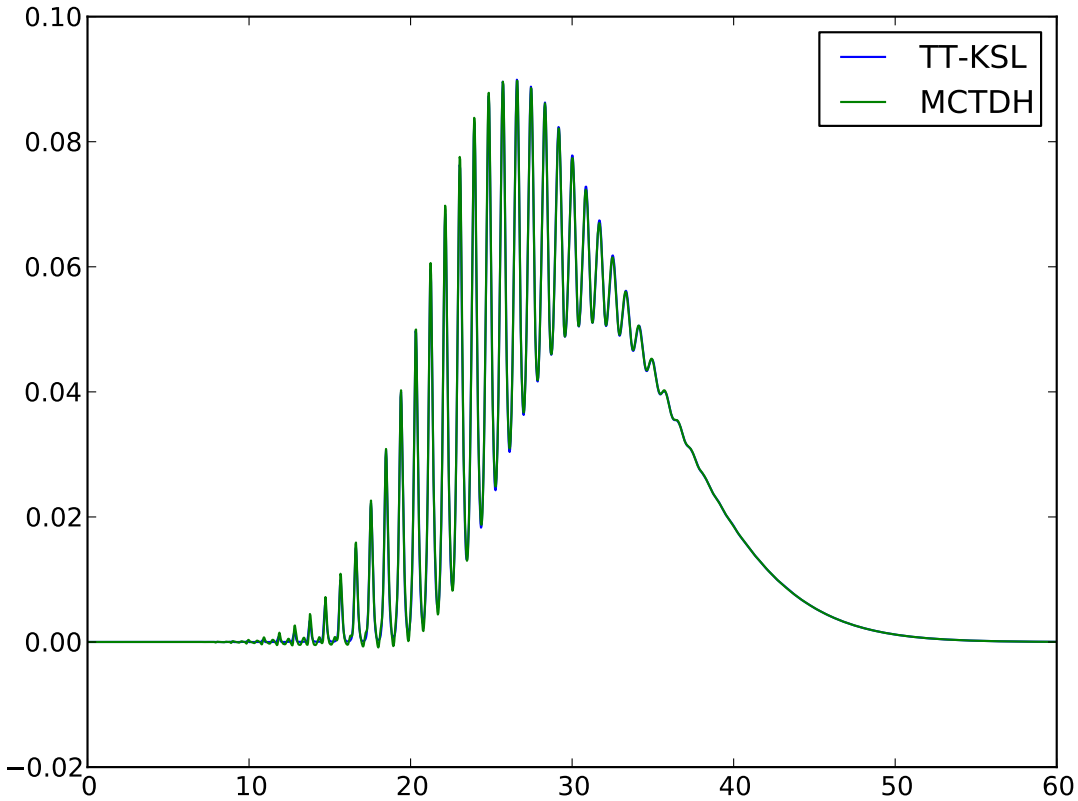
Multidimensional case

- The extension to HT/TT format is (almost done)
- The extension to $\frac{dy}{dt} = Ay$ is done
- The integrator can be readily applied to optimization problems (project the gradient/Newton step)
- Implemented in **ttpy** package (http://github.com/oseledets/ttpy)
- The CECAM workshop: C. Lubich, B. Vandereycken, R. Schneider, I. Oseledets talks

Comparison with MCTDH

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

http://www.pci.uni-heidelberg.de/cms/mctdh.html



Other techniques

There are other important techniques for non-stationary problems

DG

V. Kazeev, O. Reichmann, Ch. Schwab

Discontinious Galerkin in time:

Leads to shifted linear systems

$$(A - \lambda_i I) u_i = f$$

- DMRG/AMEN solvers, estimates
- Non-symmetric problems (can be harsh)

Block time scheme

Time as a dimension:

O., S. Dolgov, B. Khoromskij

Block time scheme

Time as a dimension:

O., S. Dolgov, B. Khoromskij

Consider an Euler scheme:

$$y_{k+1} = (I - \tau A) y_k = S y_k$$

Write as big linear system with $N \times T$ unknowns:

$$(I \otimes Z - S \otimes I) Y = f$$

Take $T = 2^d$.

Apply DMRG/AMEN!

Open questions

Several open questions

- How to adapt the rank
- "Stable" low-rank parametrization (U, S, V is bad)!
- Estimates for the projected methods

Multiparametric problems

One of the most interesting applications — solution of multiparametric problems

$$A(p)u(p) = f(p)$$
$$A(p) : \mathbb{R}^N \to \mathbb{R}^N$$
$$p = (p_1, \ldots, p_M)$$

Multiparametric problems

After discretization, we are left with a huge linear system

$$\mathbb{A}u = f,$$

where u is $N \times P \times \ldots \times P$ — a tensor!

Tensor-structured linear system

After discretization, we are left with a huge linear system

$$\mathbb{A}u = f,$$

where u is $N \times P \times \ldots \times P$ — a tensor!

Tensor-structured linear system

After discretization, we are left with a huge linear system

$$\mathbb{A}u = f,$$

where u is $N \times P \times \ldots \times P$ — a tensor!

$$A = A_0 + A_1 p_1 + A_2 p_2 + \ldots A_M p_M$$

$$\mathbb{A} = I + I \otimes D \otimes I + \ldots$$ — rank M.

Tensor-structured linear system

Matrix format, no CP requirements

$$\mathbb{A} \approx A_1(i_1, j_1) A_2(i_2, j_2) \dots A_d(i_d, j_d).$$

How to solve multiparametric problems

What about multiparametric problems?
Can not apply DMRG/AMEN directly
One mode (first one) is large

How to solve multiparametric problems

Idea is to combine with model reduction techniques (POD, Global POD)

How to solve multiparametric problems

Idea is to combine with model reduction techniques (POD, Global POD)

$$u = Uz$$

Then:

$$A(p)u(p) = f(p) \text{ -> } U^{\top}A(p)Uz(p) = f(p)$$

(Galerkin projection).

This system is much smaller, and can be solved by dmrg_solve

We do not know the basis in advance!

How to solve multiparametric problems

Maxvol-ALS iteration:

- Know the U basis -> solve the reduced problem for $z(p)$.
- Find maximal volume p^*
- Compute new snapshots at these points
- Cycle until convergence

How to solve multiparametric problems

Suppose the separation rank between x and p is r:

$$u(x,p) \approx \sum_{\alpha=1}^{r} U(x,\alpha) Z(\alpha,p)$$

The solution can be recovered from values in r physical points!

Diffusion equation

4-parameter problem

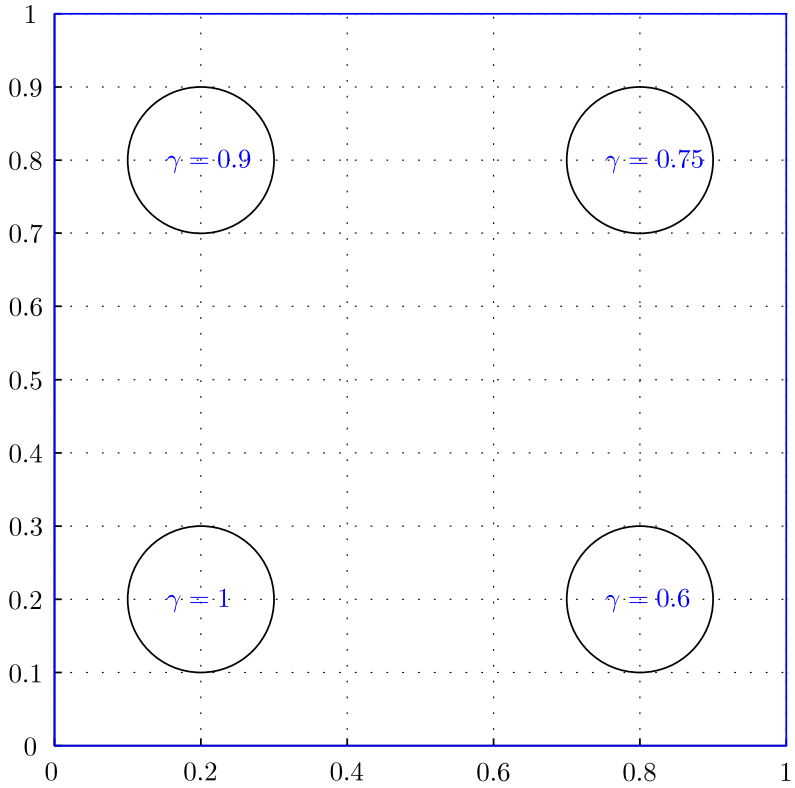
Precision: $\sim 10^{-5}$

Grid: 256×256

S. Galerkin: $\sim \times 100$ larger system

S. Collocation: ~ 1000 more solvers

DMRG + QTT: ~ 100 solves



Instead of $256^2 \times 128^4 = 2^{44}$ we have

around 50000 parameters.

Open questions

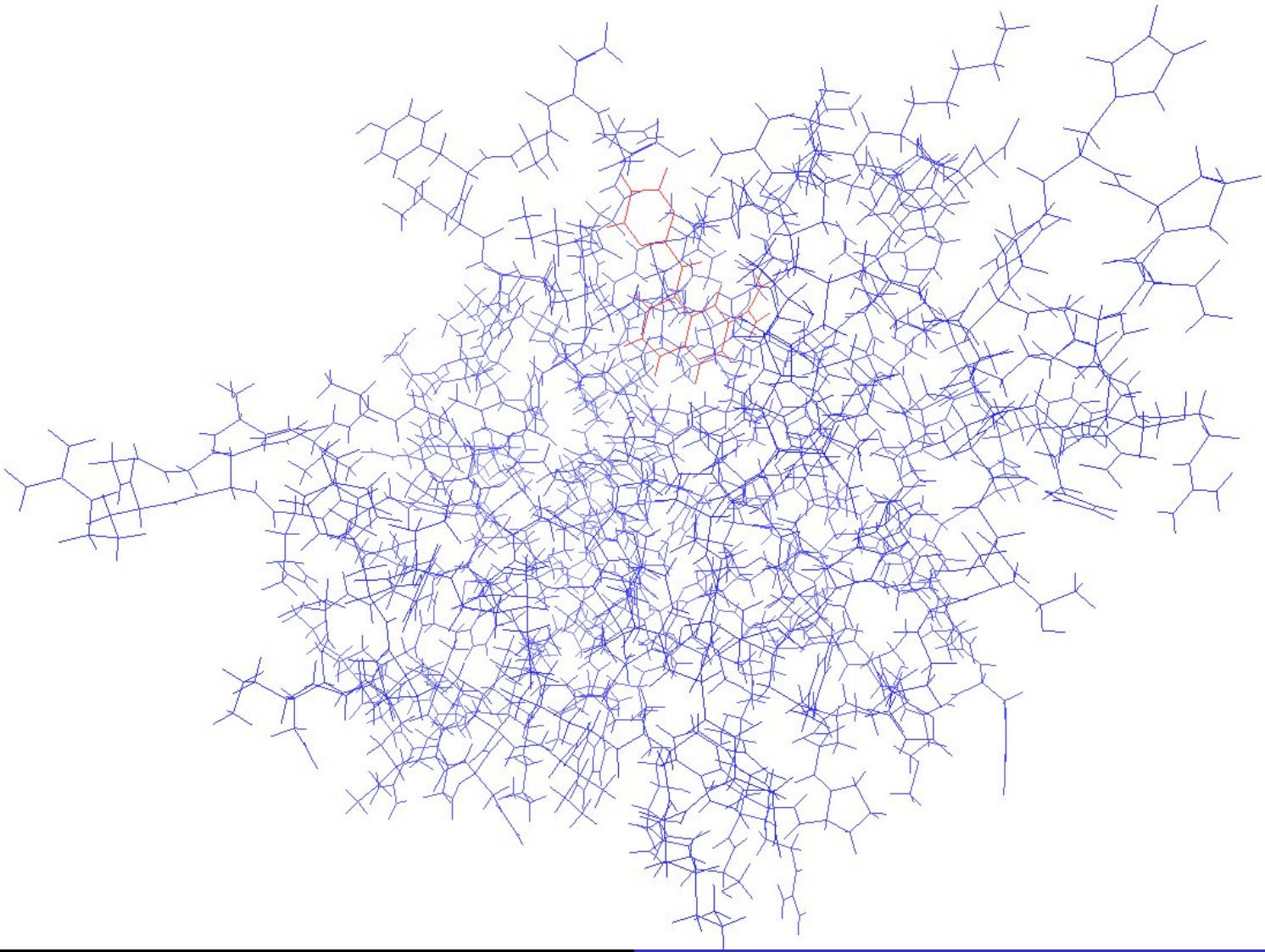
- What class of problems is tractable
- Compare with other techniques (DEIM, sparse grids)
- Efficient numerical implementations are still lacking

Problem of global optimization

It is natural to apply to the problems of global optimization

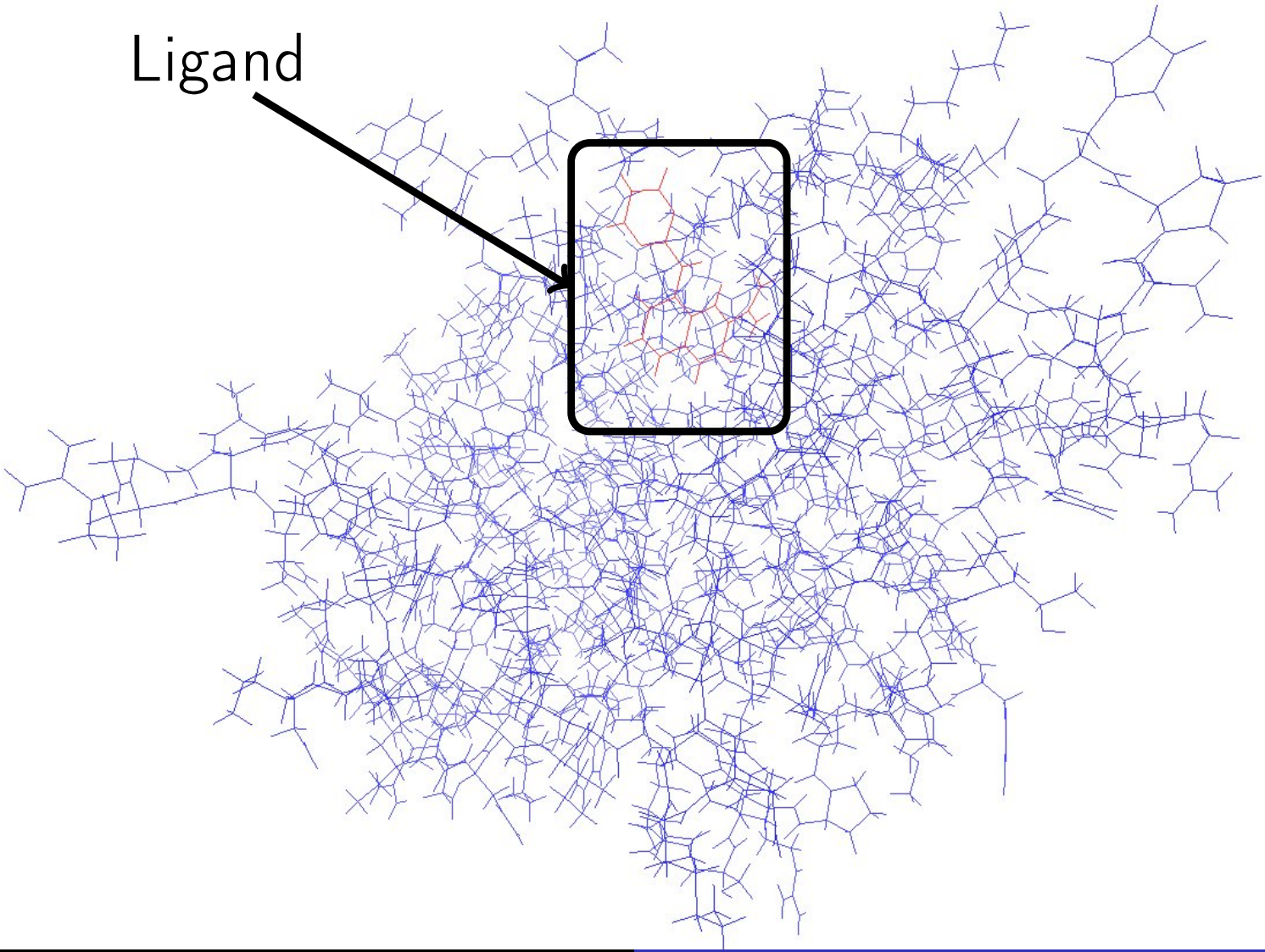
Protein-ligand

Drug design:



Protein-ligand

Drug design:



Problem statement

D. Zheltkov

Drug design, 9-variable function

Many local minima, singularities

Problem statement

D. Zheltkov

Drug design, 9-variable function

Many local minima, singularities

Need global minimum.

Problem statement

$$f(x_1, \ldots, x_d) \longrightarrow \min$$

Use cross approximation:

Problem statement

$$f(x_1, \ldots, x_d) \to \min$$

Use cross approximation:

Use magic transform: $\widehat{f} = \operatorname{arcctg} f$

Results

	Min	Max	Average
Fun calls	18804158	26272810	$2.39 * 10^7$
Min value	-65.1472	-52.7327	-62.9526
Mean val dev., Å	0.666	2.656	0.954
Mean val dev., Å	0.064	2.758	0.474

(All points — 10^9)

Results

np	8	16	32	64	128
Time, c	4834.6	2497.2	1288.1	657.93	336.12
Speedup	7.85	15.2	29.4	57.6	113

Open questions

Open questions:

- Theoretical justification (we have in two dimensions)
- Testing and applications

Latent variable models

A very interesting application:

latent variable models

Latent variable models

A very interesting application:

latent variable models

Observed variables $S_1, \ldots, S_N$ (stock prices)

There are latent variables.

Latent variable models

A very interesting application:

latent variable models

Observed variables $S_1, \ldots, S_N$ (stock prices)

There are latent variables.

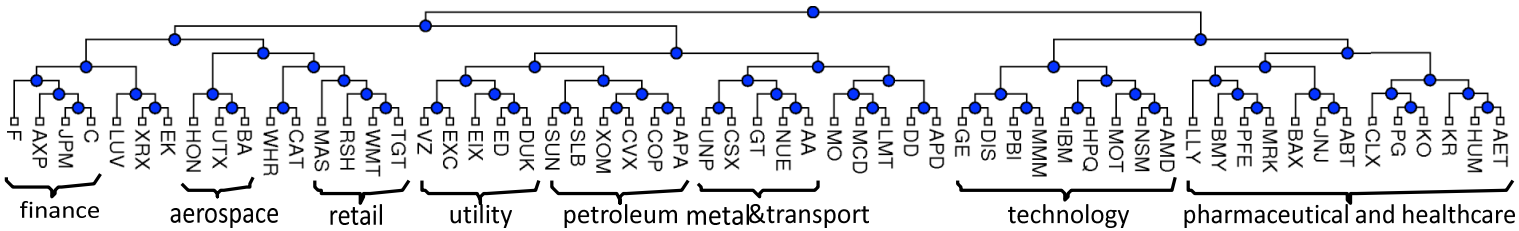
$$p(x_1, x_2) = \sum_{\alpha=1}^{r} p_1(x_1, h) w(h) p_2(x_2, h)$$

You can use tensors! (Ishteva, Le Song, Georgia Tech.)

Latent variable models

Latent tree reconstruction

(M. Ishteva, Le Song)



Open problems

- Test the effectiveness of latent tree detection techniques (Ishteva et. al work is based on a greedy algorithm, MPS people have another approach)
- Approximation of tensors with approximate entries, only through matrix-by-vector products

Hastings area law

- An area law for one dimensional quantum systems, M. B. Hastings
- An improved 1D area law for frustration-free systems, Itai Arad, Zeph Landau, Umesh Vazirani

Area law

If H is $H = \sum_{i=1} H_i$, H_i is 2-local interaction, and H has a spectral gap ε. Then, the ground-state wavefunction can be approximated in the TT (MPS) format. Entropy bound:

$$S_{1D} \leq \mathcal{O}(1) X^3 \log^8 X, \quad X = \frac{J \log d}{\varepsilon}, \quad \|H_i\| \leq J$$

Idea of the proof

The proof is linear algebra!

$H = \sum_i Q_i, \quad P_i = I - Q_i$ — projectors.

Need to estimate the TT-ranks

Idea of the proof

Idea: Find an operator K:

1. $K\psi = \psi$ for a ground-state
2. $\|K\psi_\perp\| \leq \Delta \|\psi_\perp\|$
3. $\mathbf{rank}(K\psi) \leq D\mathbf{rank}(\psi)$
4. $D\Delta \leq \frac{1}{2}$

Then, we can take any rank-1 function apply K and truncate to rank-1.

The overlap will decrease until $\sqrt{\frac{1}{2D}}$.

Idea of the proof

Constucting the operators:

$$\Pi_{even} = P_2 P_4 \ldots, \quad \Pi_{odd} = P_1 P_3 \ldots$$

$A = \Pi_{even}\Pi_{odd}$ is an approximation to the ground-state projection.

- $D_0 = d^2$
- $\Delta_0 \approx 1 - c\varepsilon$.
- $\widehat{A} = \widehat{\Pi}_m \Pi_{even}\Pi_{odd}$
- $\widehat{\Pi}_m = C_m(N)^q, \quad N = \sum_{i=1}^{m}(I - P_i)$, C_m are Chebyshev polynomials

Summary of the course

- Novel tensor formats
- High-dimensional applications (nonstandard ones!)
- Relation to Quantum Information
- A lot of open problems
- Software

Open question

- How to select variables such as they separate?