Time integration of tree tensor networks

Christian Lubich
Univ. Tübingen

Workshop on Tensor Network Algorithms, Grenoble, 7 October 2024

Outline

Numerical experiment: TTNs for a quantum spin system

Dynamical low-rank approximation: the matrix case

From matrices to tree tensor networks

Outline

Numerical experiment: TTNs for a quantum spin system

Dynamical low-rank approximation: the matrix case

From matrices to tree tensor networks

Quantum spin system

Standard example: Ising model in a transverse field

$$\mathrm{i}\,\partial_t \psi = H\psi \quad \text{with} \quad H = -\Omega \sum_{k=1}^{d} \sigma_1^{(k)} - \sum_{k=1}^{d-1} \sigma_3^{(k)} \sigma_3^{(k+1)}$$

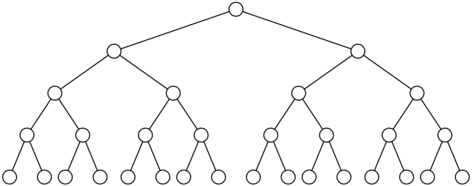
with $\Omega > 0$ and Pauli matrices $\sigma_j^{(k)}$ acting on the kth particle

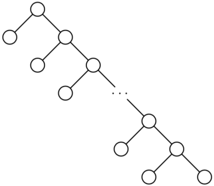
---

Approximate $\psi(t) \in \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2 \simeq \mathbb{C}^{2^d}$ by a
time-dependent tree tensor network (TTN),
possibly with adaptively chosen bond dimensions.

Different trees

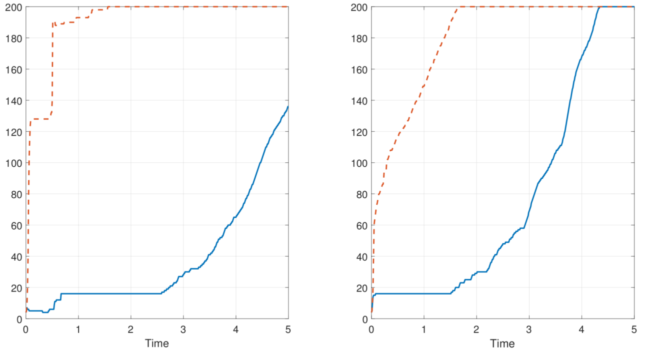
tree of minimal height (balanced tree) vs.
tree of maximal height $\rightarrow$ matrix product states





$d = 16$

Maximal bond dimensions vs. time

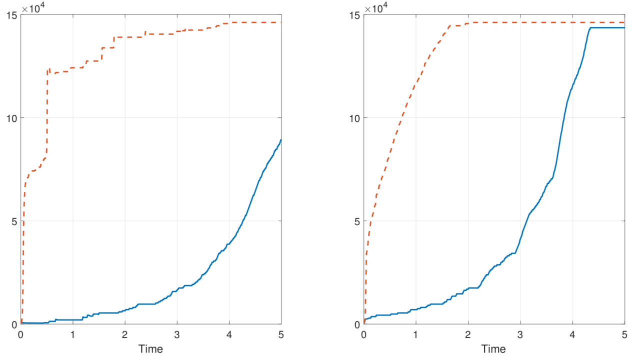


Ceruti, L., Sulz 2023, SIAM J. Numer. Anal.

---

cf. *Sulz, L., Ceruti, Lesanovsky, Carollo 2024, Phys. Rev. A*
for a long-range dissipative Ising model

Number of independent parameters vs. time



- Ising model has only nearest-neighbour interactions, which are well represented by a matrix product state (MPS).
- However, MPSs appear to struggle with capturing long-range effects for this model, as compared with TTNs on balanced trees.

Topic of this talk

What are the numerical methods
behind such TTN computations?

Topic of this talk

What are the basics of the numerical methods behind such TTN computations?

Outline

Numerical experiment: TTNs for a quantum spin system

Dynamical low-rank approximation: the matrix case

From matrices to tree tensor networks

Dynamical low-rank approximation: setting

Low-rank approximation of a matrix widely used for
data compression and model reduction

Approximate the unknown solution $A(t) \in \mathbb{C}^{m \times n}$ of a matrix ODE

$$\dot{A} = F(A)$$

by low-rank matrices: use SVD-like decomposition

$$A(t) \approx Y(t) = U(t)S(t)V(t)^*,$$

where $U(t) \in \mathbb{C}^{m \times r}$, $V(t) \in \mathbb{C}^{n \times r}$ have orthonormal columns,
$S(t) \in \mathbb{C}^{r \times r}$ is invertible.

$$\text{rank} \;\; r \ll m, n$$

Dynamical low-rank approximation

Low-rank manifold $\mathcal{M} = \{ Y \in \mathbb{C}^{m \times n} \, : \, \text{rank } Y = r \}$
Orthogonal projection onto the tangent space at $Y \in \mathcal{M}$: P_Y

Dynamical low-rank approximation: find $Y(t) \in \mathcal{M}$ from ODE

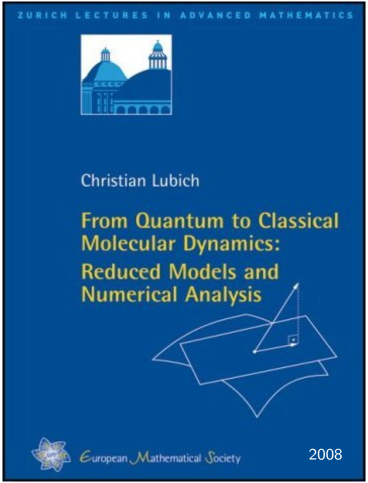
$$\dot{Y} = P_Y F(Y), \qquad Y(0) \in \mathcal{M}.$$

Project the vector field onto the tangent space of the approximation manifold

---

Dirac 1930, quantum physics: time-dependent variational principle

Tangent space projection



$$\dot{A} = F(A)$$

is approximated by

$$\dot{Y} = P_Y F(Y)$$

Dirac–Frenkel time-dependent variational principle

Differential equations for the factors

$$\dot{A} = F(A) \qquad \text{is approximated by} \qquad \dot{Y} = P_Y F(Y)$$

$$Y(t) = U(t)S(t)V(t)^T \approx A(t)$$

with

$$\begin{aligned}
\dot{U} &= (I_m - UU^T)F(Y)VS^{-1} \\
\dot{V} &= (I_n - VV^T)F(Y)^T US^{-T} \\
\dot{S} &= U^T F(Y)V
\end{aligned}$$

---

derived and analysed by *Koch & L. 2007*

Small singular values: high curvature

$\dot{Y} = P_Y F(Y)$ yields ODEs for the factors of $Y = USV^*$.

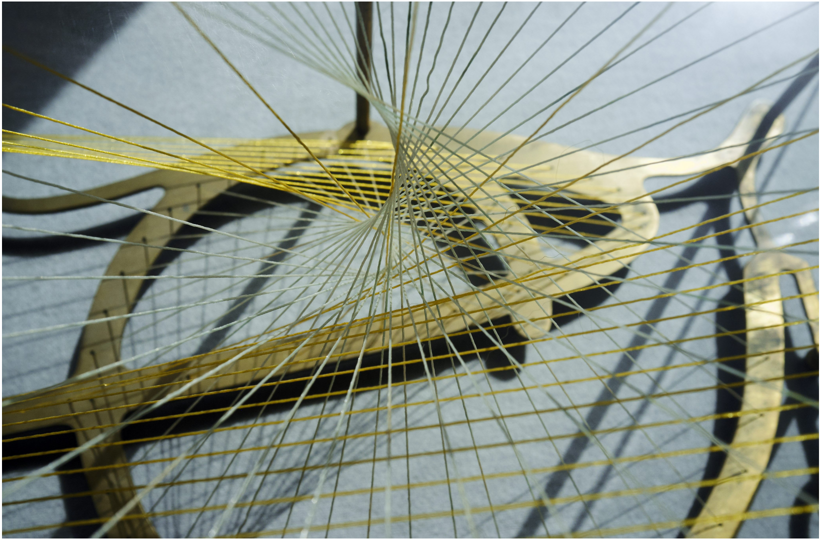
However, the ODEs for U, S, V are a pain to integrate numerically: they contain S^{-1} as factor, S is typically ill-conditioned

Geometric obstruction: with σ_r = smallest nonzero singular value,

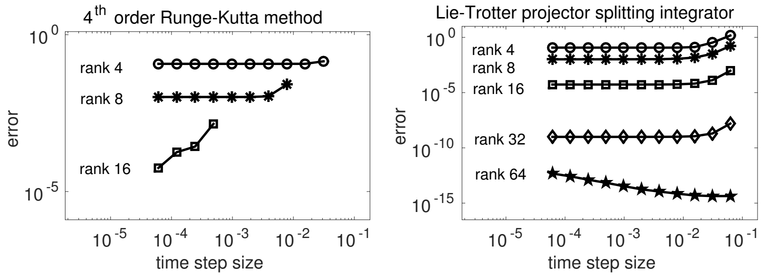
$$\frac{1}{\sigma_r} \sim \text{curvature of } \mathcal{M} \text{ at } Y$$

Is tangent space projection a reasonable approach for a manifold with high curvature?

Ruled surface



Numerical experiment: integrator errors



Runge-Kutta method (left) and projector-splitting integrator (right) for different approximation ranks and stepsizes, for a problem with singular values $2^{-j}e^t$ for $j = 1, \ldots, 100$, at $t = 1$.

Projector-splitting integrator

Split the tangent space projection, which at $Y = USV^*$ is an alternating sum of three subprojections:

$$P_Y Z = ZVV^* - UU^*ZVV^* + UU^*Z.$$

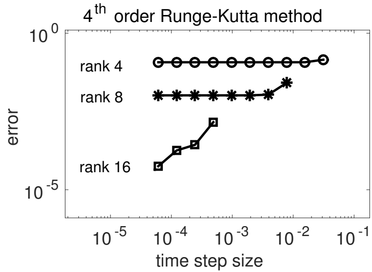
Splitting integrator:

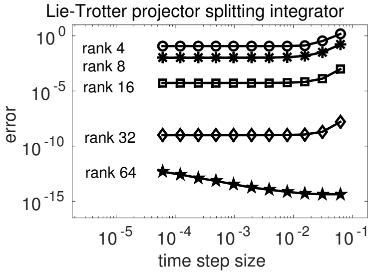
- updates the factorization $Y_n = U_n S_n V_n^*$ from n to $n+1$.
- alternates between solving differential equations for slim matrices (US, S, VS^*) and orthogonal decompositions.

L. & Oseledets 2014, *BIT*

Extension to matrix product states:
L., O. & Vandereycken 2015, *SIAM J. Numer.Anal.*
Haegeman, L., O., V., Verstraete 2016, *Phys. Rev. B*
misnomer "TDVP"

Numerical experiment: integrator errors





Runge-Kutta method (left) and projector-splitting integrator (right) for different approximation ranks and stepsizes, for a problem with singular values $2^{-j}e^{t}$ for $j = 1, \ldots, 100$, at $t = 1$.

The projector-splitting integrator is robust to small singular values.

Robustness to small singular values

The projector-splitting integrator

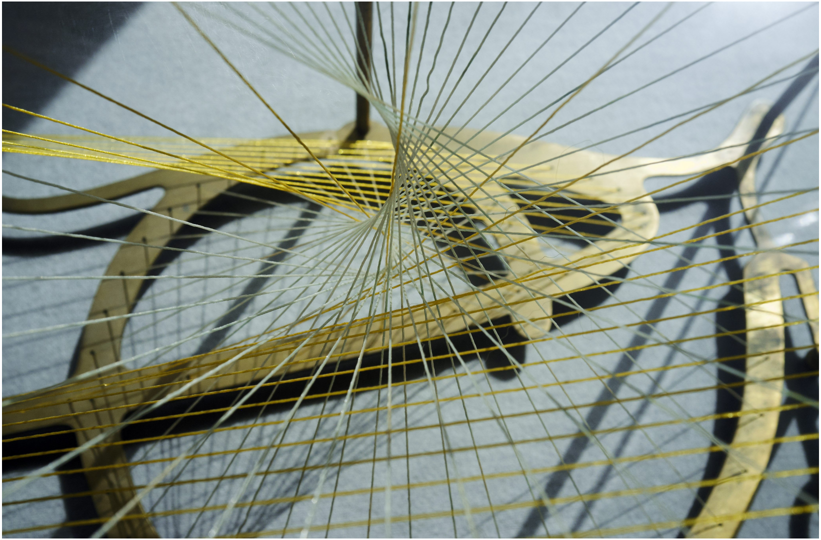
- reproduces rank-r matrices exactly.
- admits convergent error bounds that are independent of the singular values.

Why so robust?

In each substep of the algorithm, the approximation moves along a flat subspace of the manifold $\mathcal{M}$ of rank-r matrices. In this way, the high curvature due to small singular values does no harm.

Kieri, L. & Walach 2016

Ruled surface



More flexible framework: BUG integrators

Basis Update & Galerkin integrators

In a time step starting from the factored rank-r_0 matrix $U_0 S_0 V_0^*$, update to a factored rank-r_1 matrix $U_1 S_1 V_1^*$:

1. Update and augment the orth. bases U_0, V_0 to $\widehat{U}, \widehat{V}$:
   – integrate $\dot{K} = F(KV_0^*)V_0, \quad K(t_0) = K_0 = U_0 S_0$ and
   $$\dot{L} = F(U_0 L^*)^* U_0, \quad L(t_0) = L_0 = V_0 S_0^*$$
   – orthogonalise: $\widehat{U} =$orth$[K_0, K_1]$ and $\widehat{V} =$orth$[L_0, L_1]$
   ($2r_0$ basis vectors)
2. Use a variational method (Galerkin method) with the augmented bases $\widehat{U}$ and $\widehat{V}$ to update S_0 to $\widehat{S}_1 \in \mathbb{C}^{2r_0 \times 2r_0}$.
3. Truncate $\widehat{S}_1 \to S_1 \in \mathbb{C}^{r_1 \times r_1}$ and reduce bases to U_1, V_1 with r_1 basis vectors via an SVD of $\widehat{S}_1$, with adaptive rank r_1 controlled by the given truncation tolerance

Ceruti, Kusch, L. 2022 – 2024

More flexible framework: BUG integrators (ctd.)

Favourable properties:

- easy rank adaptivity and enhanced parallelism,
- conservation and dissipation properties (up to truncation),
- preserves symmetry and anti-symmetry (bosons and fermions)
- no backward time step (dissipative problems)

- fully parallel version
- variants with higher robust approximation order

again: robust to small singular values

Ceruti, Kusch, L. 2022 – 2024

Outline

Numerical experiment: TTNs for a quantum spin system

Dynamical low-rank approximation: the matrix case

From matrices to tree tensor networks

From low-rank matrices to tree tensor networks

Systematic extension of low-rank integrators and their properties:

Low-rank matrices of rank r
$\rightarrow$ Tucker tensors of multilinear rank $(r_1, \ldots, r_m)$
$\rightarrow$ general tree tensor networks (TTN) of tree rank $(r_\tau)_{\tau \leq \bar{\tau}}$

Formulation, implementation and analysis of numerical methods for TTNs require a concise common mathematical formalism. (Pictures can be helpful to develop some intuition.)

- Projector-splitting integrator for TTNs:
  Ceruti, L. & Walach 2021, *SIAM J. Numer. Anal.*
- Rank-adaptive BUG integrator for TTNs:
  Ceruti, L. & Sulz 2023, *SIAM J. Numer. Anal.*
- Parallel rank-adaptive BUG integrator for TTNs:
  Ceruti, Kusch, L. & Sulz 2025, *SIAM J. Sci. Comput.*

Outline

Numerical experiment: TTNs for a quantum spin system

Dynamical low-rank approximation: the matrix case

From matrices to tree tensor networks