

Simulation of long-range quantum spin systems using tree tensor networks

Dominik Sulz,

based on joint work with F. Carollo, G. Ceruti, I. Lesanovsky and Ch. Lubich

University of Tübingen

TRR Junior Meeting Herrsching
15.12.2023

Problem: Long-range interacting spin systems

We consider D particles undergoing Markovian open quantum dynamics. The state of the system is described by a (vectorized) density matrix $\rho(t)$ and evolves through

$$\dot{\rho}(t) = \mathcal{L}[\rho(t)] := -i[H, \rho(t)] + \mathcal{D}[\rho(t)],$$

where

$$H = \Omega \sum_{k=1}^D \sigma_x^{(k)} + \Delta \sum_{k=1}^D n^{(k)} + \sum_{k \neq h} \frac{V}{|k-h|^\alpha} n^{(k)} n^{(h)}$$
$$\mathcal{D}[\rho] = \gamma \sum_{k=1}^D \left(J^{(k)} \rho J^{(k),*} - \frac{1}{2} \left(\rho J^{(k),*} J^{(k)} + J^{(k),*} J^{(k)} \rho \right) \right).$$

Here

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad n = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad J = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Problem: Long-range interacting spin systems

We consider D particles undergoing Markovian open quantum dynamics. The state of the system is described by a (vectorized) density matrix $\rho(t)$ and evolves through

$$\dot{\rho}(t) = \mathcal{L}[\rho(t)] := -i[H, \rho(t)] + \mathcal{D}[\rho(t)],$$

where

$$H = \Omega \sum_{k=1}^D \sigma_x^{(k)} + \Delta \sum_{k=1}^D n^{(k)} + \sum_{k \neq h} \frac{V}{|k-h|^\alpha} n^{(k)} n^{(h)}$$
$$\mathcal{D}[\rho] = \gamma \sum_{k=1}^D \left(J^{(k)} \rho J^{(k),*} - \frac{1}{2} \left(\rho J^{(k),*} J^{(k)} + J^{(k),*} J^{(k)} \rho \right) \right).$$

Memory requirement full problem:

$D = 8 \rightarrow 500KB$

$D = 16 \rightarrow 34GB$

$D = 32 \rightarrow 147EB (= \text{exabytes})$

Problem: Long-range interacting spin systems

Consider a general tensor differential equation

$$\dot{A}(t) = F(t, A(t)), \quad A(t_0) = A_0.$$

For example $F = \mathcal{L}$ from the slide before.

Problem: Dimension of the Hilbertspace grows exponentially with the number of particles 2^D (closed) or 4^D (open).

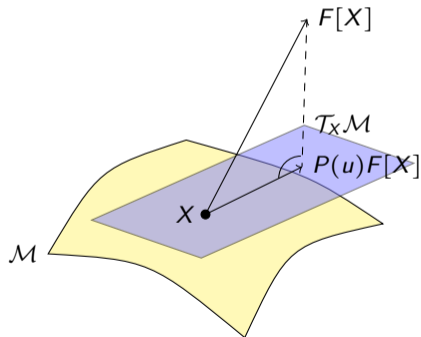
Idea: Project the equation to a low-dimensional manifold \mathcal{M} . I.e. solve

$$\dot{Y}(t) = P(Y)F(t, Y(t)), \quad Y(t_0) = Y_0 \in \mathcal{M}.$$

Dynamical low rank approximation

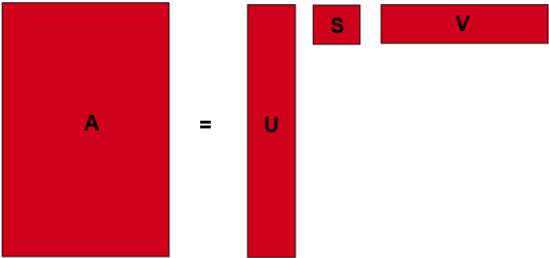
On a manifold \mathcal{M} we impose the time-dependent Dirac-Frenkel variational principle. We determine $X = X(t)$ such that its derivative \dot{X} , which lies in $\mathcal{T}_X\mathcal{M}$, satisfies

$$\dot{X} \in \mathcal{T}_X\mathcal{M} \text{ such that } \langle \dot{X} - F[X], Y \rangle = 0 \quad \forall Y \in \mathcal{T}_X\mathcal{M}.$$



Dynamical low rank approximation

Consider first the simplest (matrix) case: Let \mathcal{M} be the manifold of matrices of rank r . A low-rank decomposition of $A \in \mathcal{M}$ corresponds to the singular value decomposition of a matrix $A \in \mathbb{R}^{n \times m}$:

$$A = USV^T$$


The memory footprint reduces from nm to $nr + mr + r^2$.

→ **Idea:** Keep the decomposed structure over the time integration!

Dynamical low rank approximation

Make the ansatz $\psi(t) = U(t)S(t)V(t)^\top$. Inserting this in the Dirac-Frenkel variational principle,

$$\langle \dot{\psi}(t) - F(\psi(t)), Y \rangle = 0 \quad \forall Y \in \mathcal{T}_\psi \mathcal{M},$$

we obtain.

Equation of motion for the factors:

$$\dot{U}(t) = (I - U(t)U(t)^\top)F(\psi(t))V(t)S(t)^{-1}$$

$$\dot{S}(t) = U(t)^\top F(\psi(t))V(t)$$

$$\dot{V}(t) = (I - V(t)V(t)^\top)F(\psi(t))U(t)S(t)^{-\top}$$

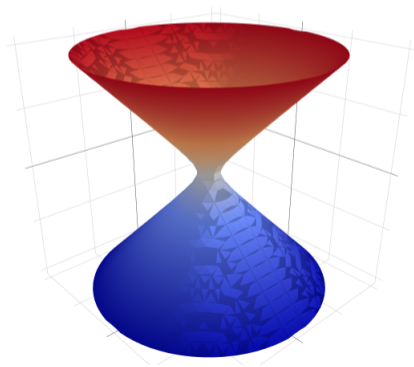
Dynamical low rank approximation

$$\dot{U}(t) = (I - U(t)U(t)^\top)F(\psi(t))V(t)S(t)^{-1}$$

$$\dot{S}(t) = U(t)^\top F(\psi(t))V(t)$$

$$\dot{V}(t) = (I - V(t)V(t)^\top)F(\psi(t))U(t)S(t)^{-\top}$$

Problem: The inverses of $S(t)$ can get close to singular i.e. we have a high curvature of the manifold.



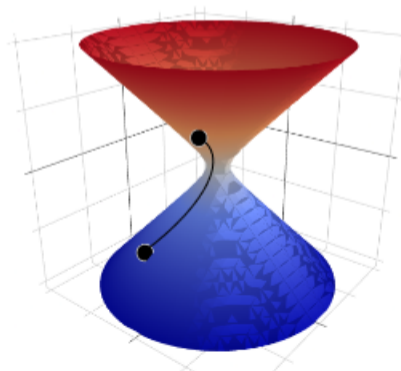
Dynamical low rank approximation

$$\dot{U}(t) = (I - U(t)U(t)^\top)F(\psi(t))V(t)S(t)^{-1}$$

$$\dot{S}(t) = U(t)^\top F(\psi(t))V(t)$$

$$\dot{V}(t) = (I - V(t)V(t)^\top)F(\psi(t))U(t)S(t)^{-\top}$$

Problem: The inverses of $S(t)$ can get close to singular i.e. we have a high curvature of the manifold.



Robust time integration

We see that we need robust time integration schemes which are insensitive in the presence of small singular values. Several methods have been derived in the last years:

- Projector splitting integrator
- Basis update Galerkin (BUG)
- Rank-adaptive BUG
- Parallel BUG

Ch. Lubich, I. Oseledets, BIT 54 (2014), 171-188

G. Ceruti, Ch. Lubich, BIT Numer. Math. 62 (2022), 23-44

G. Ceruti, J. Kusch, Ch. Lubich, BIT Numer. Math. 62 (2022), 1149-1174

G. Ceruti, J. Kusch, Ch. Lubich, arXiv2304.05660 April 2023

BUG: Basis update and Galerkin

Input: $U_0 \in \mathbb{R}^{n \times r}$, $V_0 \in \mathbb{R}^{m \times r}$ and $S_0 \in \mathbb{R}^{r \times r}$.

Basis update: We solve the small matrix differential equations:

$$\begin{aligned}\dot{K}(t) &= F(t, K(t)V_0^T)V_0, & K(t_0) &= U_0S_0 \\ \dot{L}(t) &= F(t, U_0L(t)^T)U_0, & L(t_0) &= V_0S_0^T.\end{aligned}$$

Define $U_1 = \text{orth}(K(t_1))$ and $V_1 = \text{orth}(L(t_1))$.

Galerkin step: Solve the $r \times r$ differential equation

$$\dot{S}(t) = U_1^T F(t, U_1S(t)V_1^T)V_1, \quad S(t_0) = S_0.$$

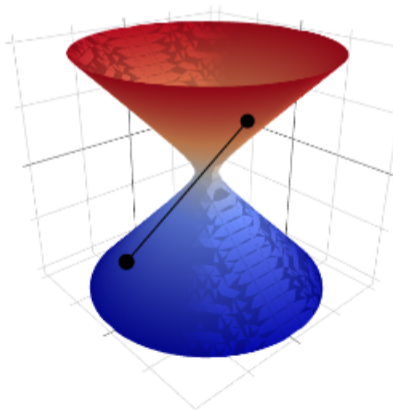
Define $S_1 = S(t_1)$.

Output: Approximation at new time step t_1 is now $U_1S_1V_1^T$.

BUG: Basis update and Galerkin

Advantages:

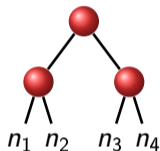
- The integration keeps the decomposed form. → No need to compute the full matrix at any time.
- Curvature of the manifold is not seen on (flat) subspaces $U(t)S(t)V_0^T$, $U_0S(t)V(t)^T$ and $U_1S(t)V_1^T$.



Tree tensor networks

Tree tensor networks (TTNs) are the natural generalization of matrices to high-dimensional tensors. A TTN is a hierarchical data sparse format to store high-order tensors. U_τ on the lowest level is a basis matrix and on higher levels it is defined recursively by

$$X_\tau = C_\tau \times_{i=1}^m U_{\tau_i}$$
$$U_\tau = \text{Mat}_i(X_\tau)^\top.$$



Note: Matrices are also TTNs via

$$A = USV^\top = S \times_1 U \times_2 V.$$



Tree tensor networks

We are now working on the manifold of TTNs with ranks $(r_\tau)_{\tau \leq \bar{\tau}}$.

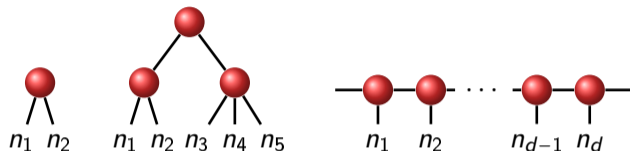


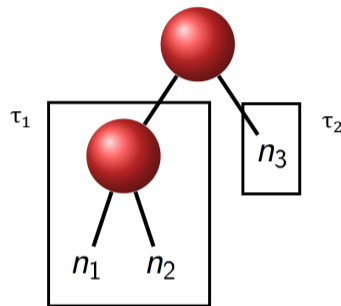
Figure: Different examples for TTN's (from left to right): Matrix, general TTN, matrix product state/tensor train.

The red balls encode a connecting tensor of matching order, which was the S matrix in the matrix case. The nodes n_i encode the basis matrices, which correspond to U and V .

(Rank-adaptive) BUG-integrator for tree tensor networks

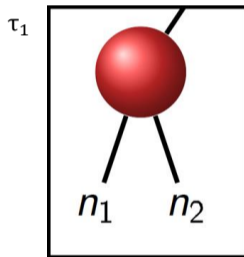
Suppose we have a TTN $X_{\bar{\tau}}$. We upload the subtrees of $X_{\bar{\tau}}$ recursively with the subflows $\Phi_{\tau}^{(i)}$ and the cores with the subflow Ψ_{τ} . For the subflow $\Phi_{\tau}^{(i)}$ there are two cases:

- The i th subtree is a leaf. Then $\Phi_{\tau}^{(i)}$ solves a small matrix ODE.
- The i th subtree is again a TTN. Then we apply the algorithm recursively to this smaller tree.



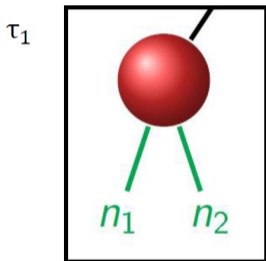
(Rank-adaptive) BUG-integrator for tree tensor networks

Now all subtrees of τ_1 are leaves. We update the leaves by solving a small matrix ODE for one time step. The solution at time δt is taken as the updated leaf.



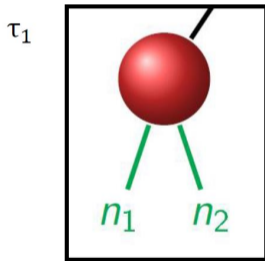
(Rank-adaptive) BUG-integrator for tree tensor networks

Now all subtrees of τ_1 are leaves. We update the leaves by solving a small matrix ODE for one time step. The solution at time δt is taken as the updated leaf.



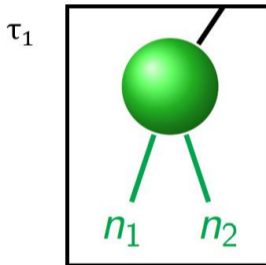
(Rank-adaptive) BUG-integrator for tree tensor networks

Now we need to update the connecting tensor $C_{\tau_1}^0$ by the subflow Ψ_{τ_i} . This is done by solving a small tensor ODE for one time step in the new basis. The solution at time δt is taken as the new connecting tensor $C_{\tau_1}^1$.



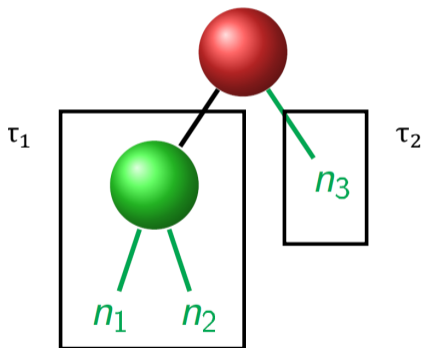
(Rank-adaptive) BUG-integrator for tree tensor networks

Now we need to update the connecting tensor $C_{\tau_1}^0$ by the subflow Ψ_{τ_i} . This is done by solving a small tensor ODE for one time step in the new basis. The solution at time δt is taken as the new connecting tensor $C_{\tau_1}^1$.



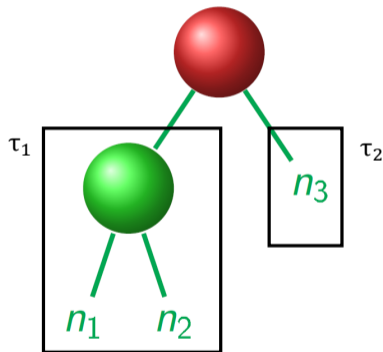
(Rank-adaptive) BUG-integrator for tree tensor networks

Remain to update the 0-dimension of C_{τ_1} by an orthogonalization in the corresponding mode.



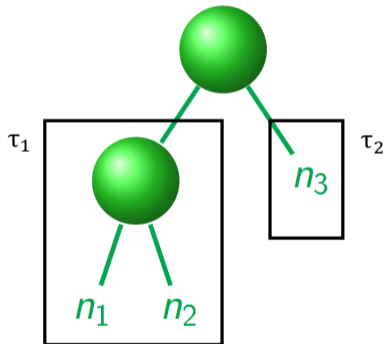
(Rank-adaptive) BUG-integrator for tree tensor networks

Remain to update the 0-dimension of C_{τ_1} by an orthogonalization in the corresponding mode.



(Rank-adaptive) BUG-integrator for tree tensor networks

And update the last root tensor C_τ by the procedure Ψ_τ , which we have seen before.



Properties

BUG integrator:

- Let $A(t)$ be the exact and $X_{\bar{\tau}}^n$ the numerical solution at time $t_0 + nh$. Further let $F_{\bar{\tau}}$ be Lipschitz continuous and bounded. Suppose that $\|(I - P(Y))F_{\bar{\tau}}(t, Y)\| \leq \epsilon \forall Y \in \mathcal{M}$ in a neighborhood of $A(t_n)$, where $P(Y)$ denotes the projection onto $\mathcal{T}_Y \mathcal{M}$. Then it holds

$$\|A(t_n) - X_{\bar{\tau}}^n\| = \mathcal{O}(h + \epsilon),$$

independent of the singular values.

Rank-adaptive BUG integrator:

Properties

BUG integrator:

- Let $A(t)$ be the exact and $X_{\bar{\tau}}^n$ the numerical solution at time $t_0 + nh$. Further let $F_{\bar{\tau}}$ be Lipschitz continuous and bounded. Suppose that $\|(I - P(Y))F_{\bar{\tau}}(t, Y)\| \leq \epsilon \forall Y \in \mathcal{M}$ in a neighborhood of $A(t_n)$, where $P(Y)$ denotes the projection onto $\mathcal{T}_Y \mathcal{M}$. Then it holds

$$\|A(t_n) - X_{\bar{\tau}}^n\| = \mathcal{O}(h + \epsilon),$$

independent of the singular values.

Rank-adaptive BUG integrator:

- If $F_{\bar{\tau}}$ satisfies $\text{Re}\langle Y, F_{\bar{\tau}}(t, Y) \rangle = 0 \forall Y$ and ϑ is the truncation tolerance it holds

$$\left| \|X_{\bar{\tau}}^1\| - \|X_{\bar{\tau}}^0\| \right| \leq c_{\bar{\tau}} \vartheta.$$

- Consider the tensor Schrödinger equation and let $E(Y) = \langle Y, H[Y] \rangle$. Then it holds for every step size h

$$|E(X_{\bar{\tau}}^1) - E(X_{\bar{\tau}}^0)| \leq c_{\bar{\tau}} \vartheta \|H[X_{\bar{\tau}}^1 + \widehat{X}_{\bar{\tau}}^1]\|.$$

Application to long-range open quantum systems

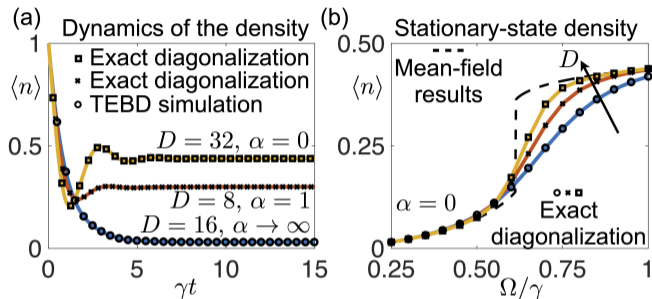
We consider D particles undergoing Markovian open quantum dynamics. The state of the system is described by a density matrix $\rho(t)$ and evolves through

$$\dot{\rho}(t) = \mathcal{L}[\rho(t)] := -i[H, \rho(t)] + \mathcal{D}[\rho(t)],$$

where

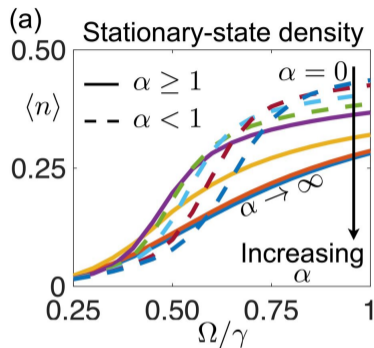
$$H = \Omega \sum_{k=1}^D \sigma_x^{(k)} + \Delta \sum_{k=1}^D n^{(k)} + \sum_{k \neq h} \frac{V}{|k-h|^\alpha} n^{(k)} n^{(h)}$$
$$\mathcal{D}[\rho] = \gamma \sum_{k=1}^D \left(J^{(k)} \rho J^{(k),*} - \frac{1}{2} \left(\rho J^{(k),*} J^{(k)} + J^{(k),*} J^{(k)} \rho \right) \right).$$

Application to long-range open quantum systems



We see that the integrator almost exactly coincides with the expected results (left). Further, we observe that for $\alpha = 0$ we indeed converge to the mean-field limit (which is known to have a phase transition) when increasing the number of particles.

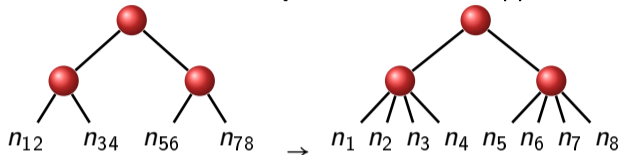
Application to long-range open quantum systems



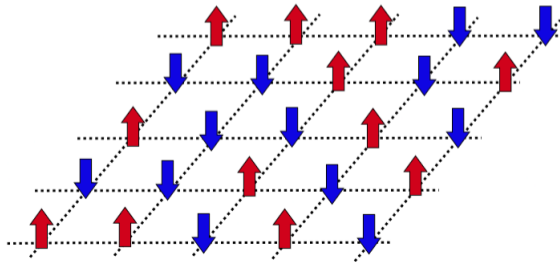
Of physical interest is whether we observe a phase transition persists for other values of α than zero. The plot above (all curves for $D = 16$) indicates that the phase transition indeed persists as $\alpha < 1$.

Current work

- Use different tree structures than binary trees \rightarrow better approximation properties?



- Generalize the parallel BUG to TTNs. \rightarrow rougher but faster time integration.
- Apply tree tensor network integrators to 2-D problems.



Thanks for your attention!