Numerical approaches to quantum many-body non-equilibrium

Université de Strasbourg

cnrs dépasser les frontières

CESQ CENTRE EUROPÉEN DE SCIENCES QUANTIQUES

www.schachenmayer.fr

www.cesq.eu

Johannes Schachenmayer: **schachenmayer@unistra.fr**

Numerical approaches to quantum many-body non-equilibrium

Goal:

A tour through some numerical methods for simulating large quantum many-body non-equilibrium dynamics, with examples. Learning physics by simulating it.

Lecture 1: Foundations (QM on a computer), Runge-Kutta, Applications to ultra-cold bosonic systems

Lecture 2: Spin-model physics, Krylov space approaches, Open system methods

Lecture 3: Large systems: Matrix Product States (DMRG), Applications to spin-models and Bose-Hubbard

- Some text recommendations:

  - *Numerical recipes - The Art of Scientific Computing (a classic), on-line: https://numerical.recipes*
  - *General references to openly available publications in class*

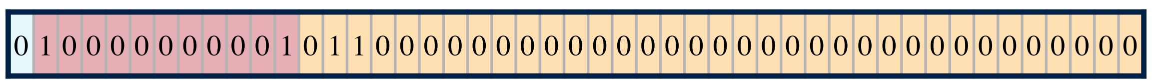
- Language recommendation (used for examples): Julia, **https://julialang.org/** (open source, easy, fast linear algebra)

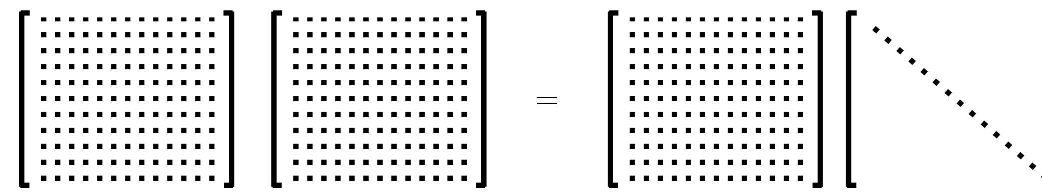
- What these lectures are **not:**

  - *Complete: Many techniques are not discussed (e.g. Monte-Carlo, Fermions, Phase space methods, …)*
  - *Computer science class: No proofs of complexity etc.*
  - *Numerical tutorial: There will be code snippets … incentive to do it yourself*

Last Time

- We discussed how to represent numbers as bits. We describe systems with state-vectors (not only in QM). If the problem is linear (QM), an exact diagonalization of the Hamiltonian solves everything, and is easily doable numerically for small systems.



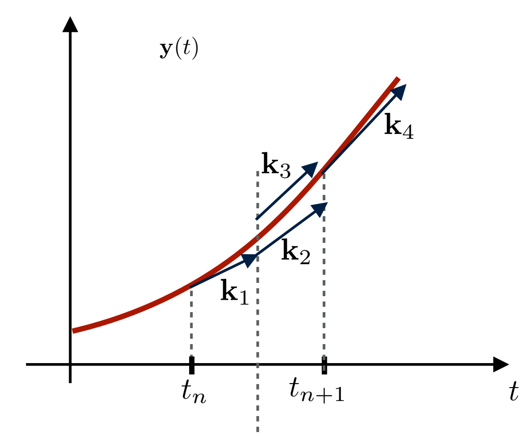


- Not all problems are linear. Also in quantum mechanics, when making a mean-field approximation (e.g. Gross-Pitaevskii equations for ultra-cold bosonic gases) the problem becomes non-linear. However the state-space drastically decreases in this case (from exponential to linear growth with system size).

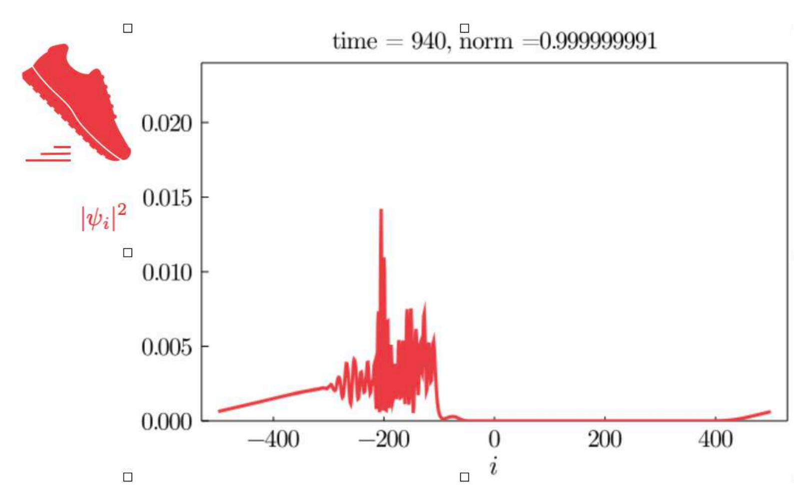
$$\frac{d}{dt}\psi(x,t) = -\mathrm{i}\left(\frac{\hat{p}^2}{2m} + V(x) + g|\psi(x,t)|^2\right)\psi(x,t)$$

- Runge-Kutta methods are a general tool to simulate dynamics of linear and non-linear problems. Formally derived from Taylor expansions using multiple steps. In particular the 4-th order method is a good compromise (stable, small error for reasonable time-step).

$$\mathbf{k}_1 = f(t_n, \mathbf{y}_n)$$
$$\mathbf{k}_2 = f\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1\right)$$
$$\mathbf{k}_3 = f\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2\right)$$
$$\mathbf{k}_4 = f\left(t_n + h, \mathbf{y}_n + h\mathbf{k}_3\right)$$
$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

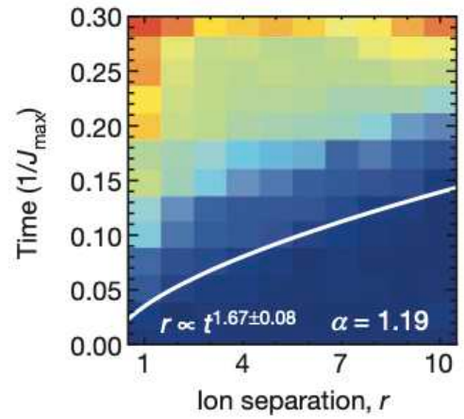


- We looked at a kicked bosonic condensate as example application, using the GP approximation, and re-produced the effect of an dynamics instability with a few lines of codes. However, this physics is actually wrong in 1D as we will see later. Mean-field approximations can drastically increase the treatable system size but are also strong approximations neglecting any entanglement.

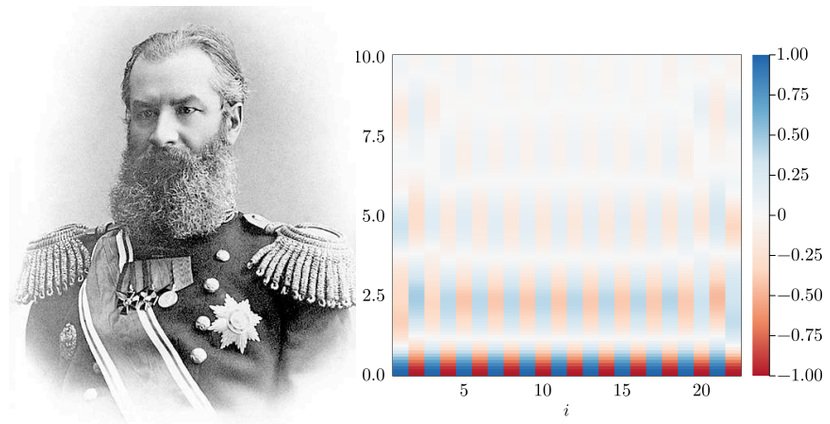


Plan for today

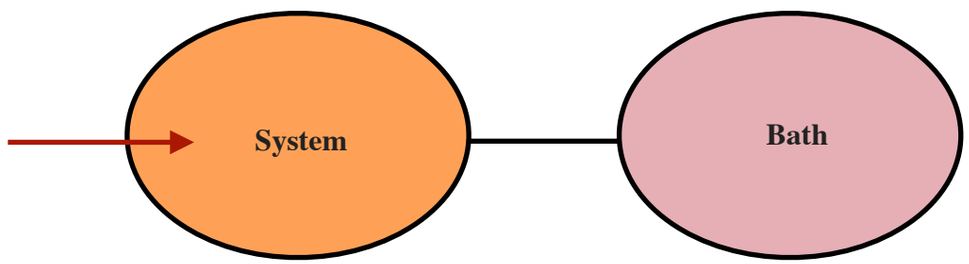
- **Part 2.1:** The many-body problem of the day: Long-range spin model dynamics. How to numerically construct Hamiltonians.



- **Part 2.2:** Simulation of dynamics with sparse Hamiltonians: Krylov space methods. Applications to spin-model dynamics.



- **Part 2.3:** Methods for open system dynamics: Density matrix linearization and quantum trajectories.



Lecture 2 - Motivation: Setup of the day

- Experiments implementing long-range spin-models effectively (e.g. with trapped ions)

$$J_{ij} = \frac{J}{|i-j|^{\alpha}}$$

Porras, D. & Cirac, J. I., Phys. Rev. Lett. 92, 207901 (2004)

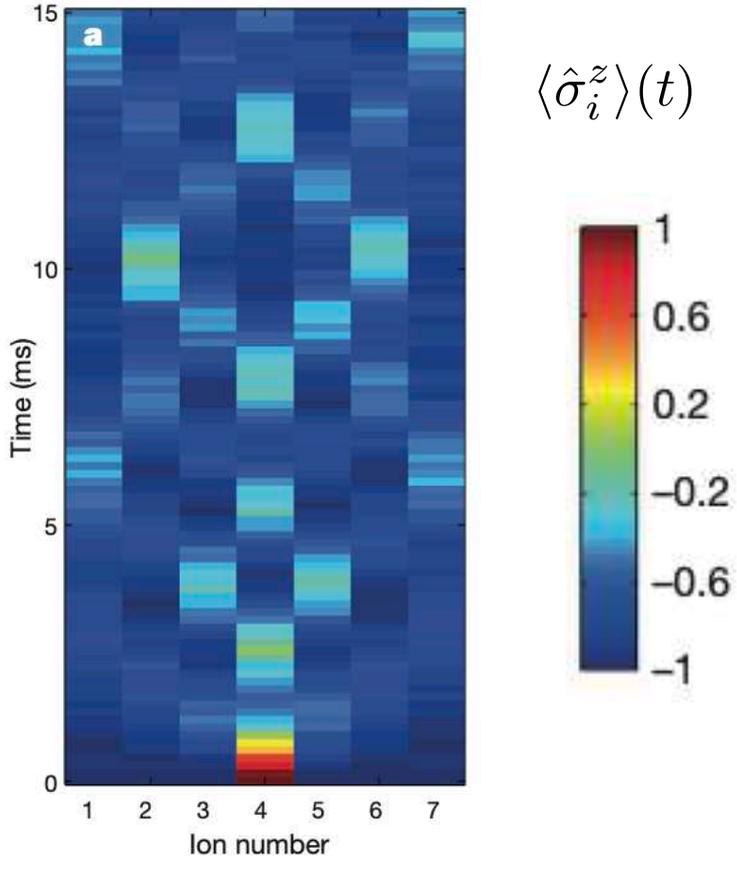
$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x$$

Long-range transverse Ising model

P. Jurcevic, et al. Nature 511, 202 (2014)
Innsbruck

$$|\psi(t=0)\rangle = |\downarrow\ldots\downarrow\uparrow\downarrow\ldots\downarrow\rangle$$

$$\langle\hat{\sigma}_i^z\rangle(t)$$



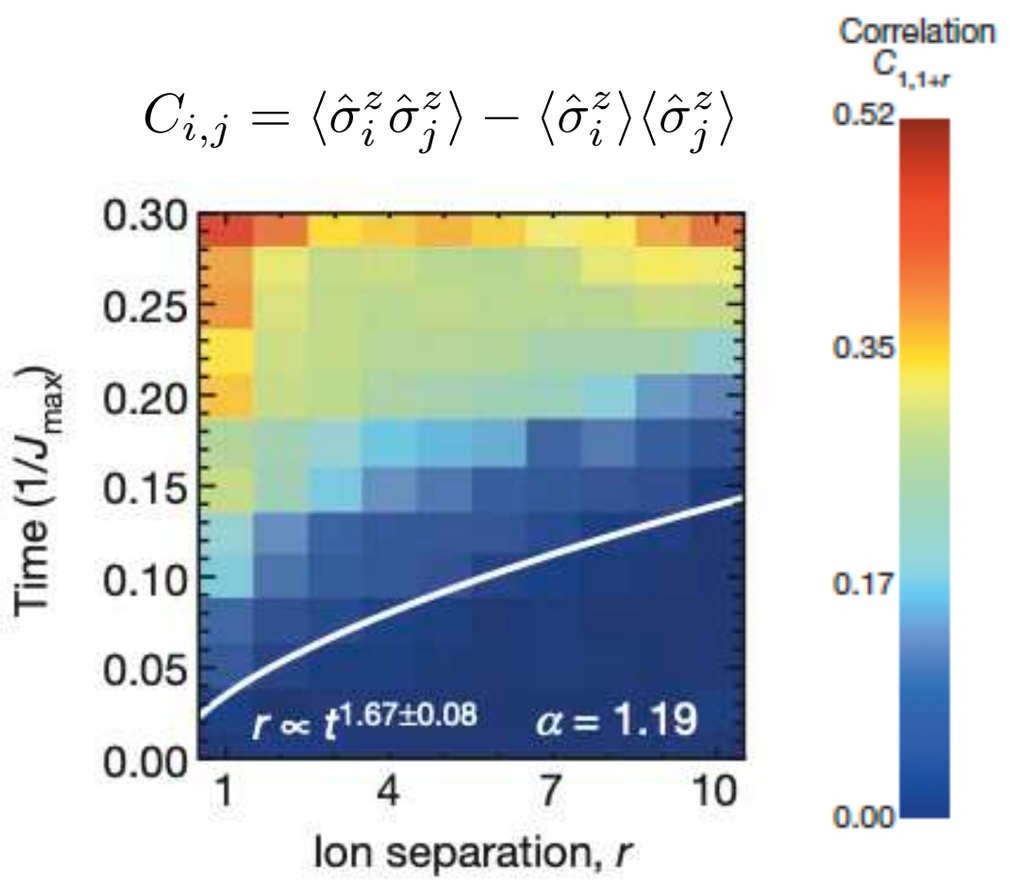
$$\hat{H}_{\mathrm{XY}} = \sum_{i<j} J_{i,j}\left(\hat{\sigma}_i^x\hat{\sigma}_j^x + \hat{\sigma}_i^y\hat{\sigma}_j^y\right) = 2\sum_{i<j} J_{i,j}\left(\hat{\sigma}_i^+\hat{\sigma}_j^- + \hat{\sigma}_i^-\hat{\sigma}_j^+\right)$$

Long-range XY model

P. Richerme, et al. Nature 511, 198 (2014)
JQI Maryland

$$|\psi(t=0)\rangle = |\downarrow\downarrow\ldots\downarrow\rangle$$

$$C_{i,j} = \langle\hat{\sigma}_i^z\hat{\sigma}_j^z\rangle - \langle\hat{\sigma}_i^z\rangle\langle\hat{\sigma}_j^z\rangle$$



Lecture 2 - Reminder Linear algebra of quantum mechanics

State = Vector

$$\begin{bmatrix} \vdots \\ \psi_i \\ \vdots \end{bmatrix} \quad D \times 1$$

In general: Complex elements

$$D \times (2 \times 64)\,\text{Bits} = D \times 16\,\text{Bytes}$$

Operators = Matrix

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ \cdots & h_{i,j} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix} \quad D \times D$$

Hamiltonian, Observables, Time-evolution operator

- Let's define a state-vector as a general concept (not limited to linear quantum mechanics)

$$\mathbf{y}(t) \begin{bmatrix} \vdots \end{bmatrix}$$

Ordinary differential equation

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t))$$

Schrödinger equation

$$\frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle$$

$$\mathbf{y} = |\psi\rangle \qquad f(t, \mathbf{y}(t)) = \hat{H}\,|\psi\rangle \qquad \textit{(linear)}$$

- The state-vector can be anything:

Linearized density matrix

$$\mathbf{y} = [\rho_{1,1}, \rho_{1,2}, \rho_{2,1}, \rho_{2,2}]^T$$

Two classical particles

$$\mathbf{y} = [x_1, p_1, x_2, p_2]^T$$

$\ldots$

Lecture 2 - Hilbert space construction in a many-body system

- In a many-body system, how do we construct Hamiltonian matrices for such spin model?

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad \textit{Transverse Ising model}$$

- What Hamiltonian terms actually mean: All terms are matrices: $2^N \times 2^N$

$$\hat{\sigma}_i^x = \cdots \otimes \mathbb{1} \otimes \sigma^x \otimes \mathbb{1} \otimes \cdots = \cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \cdots$$

spin #i *spin #i*

"tensor product" = "Kronecker product"

$$\hat{\sigma}_i^z\hat{\sigma}_j^z = \cdots \otimes \mathbb{1} \otimes \sigma^z \otimes \mathbb{1} \otimes \mathbb{1} \otimes \hat{\sigma}^z \otimes \mathbb{1} \otimes \cdots =$$

$$\cdots \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \cdots$$

spin #i *spin #j*

- Then, e.g. easy to construct a spin-lowering for spin *i:*

$$|\downarrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|\uparrow\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\hat{\sigma}^- = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\hat{\sigma}_i^- = \underbrace{\begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{pmatrix}}_{2^{i-1} \times 2^{i-1}} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \underbrace{\begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{pmatrix}}_{2^{N-i} \times 2^{N-i}}$$

Lecture 2 - Hilbert space construction in a many-body system

- In a many-body system we deal with Hamiltonians like (here: spin-model)

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j} \hat{\sigma}_i^z \hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad \textit{Transverse Ising model}$$

- Then, e.g. easy to construct a spin-lowering for spin i:

$$|\downarrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |\uparrow\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \hat{\sigma}^- = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\hat{\sigma}_i^- = \underbrace{\begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{pmatrix}}_{2^{i-1} \times 2^{i-1}} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \underbrace{\begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{pmatrix}}_{2^{N-i} \times 2^{N-i}}$$

```
# Construct lowering operators
N = 10
sm = [0 1; 0 0]
sms = Vector{}(undef, N)

for ii = 1:N
    ldim = 2^(ii-1)
    rdim = 2^(N - ii)
    sms[ii] = kron(Matrix(I, ldim, ldim), sm, Matrix(I, rdim, rdim))
end
```

Vector of lowering operators, all that's needed!

Lecture 2 - Hilbert space construction in a many-body system

- In a many-body system we deal with Hamiltonians like (here: spin-model)

$$\hat{H}_{\mathrm{TI}} = \sum_{i,j} J_{i,j} \hat{\sigma}_i^z \hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad \textit{Transverse Ising model}$$

- Then, full Hamiltonian:

```
# Construct Hamiltonian
Jij = rand(N,N) # random couplings
hx = 1;

H = zeros(2^N, 2^N)
for ii = 1:N
    H += hx .* (sms[ii]' + sms[ii]) # sx
    szii = (sms[ii]'*sms[ii] - sms[ii]*sms[ii]') # sz spin ii
    for jj = (ii+1):N
        szjj = (sms[jj]'*sms[jj] - sms[jj]*sms[jj]') # sz spin jj
        H += Jij[ii,jj] .* szii * szjj
    end
end
```

$$\hat{\sigma}_i^x = \hat{\sigma}_i^- + \hat{\sigma}_i^+$$

$$\hat{\sigma}_i^z = \hat{\sigma}_i^+ \hat{\sigma}_i^- - \hat{\sigma}_i^- \hat{\sigma}_i^+$$

- **Problem:** Inefficient!

Lecture 2 - Sparse Matrices!

- In a many-body system we deal with Hamiltonians like (here: spin-model)

$$\hat{H}_{\mathrm{TI}} = \sum_{i,j} J_{i,j} \hat{\sigma}_i^z \hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad \textit{Transverse Ising model}$$

$$N = 10$$

```
1024×1024 Matrix{Float64}:
 21.607   1.0      1.0      0.0      …  0.0      0.0      0.0      0.0
  1.0    11.6526   0.0      1.0         0.0      0.0      0.0      0.0
  1.0     0.0     14.9075   1.0         0.0      0.0      0.0      0.0
  0.0     1.0      1.0      5.81673     0.0      0.0      0.0      0.0
  1.0     0.0      0.0      0.0         0.0      0.0      0.0      0.0
  0.0     1.0      0.0      0.0      …  0.0      0.0      0.0      0.0
  0.0     0.0      1.0      0.0         0.0      0.0      0.0      0.0
  0.0     0.0      0.0      1.0         0.0      0.0      0.0      0.0
  1.0     0.0      0.0      0.0         0.0      0.0      0.0      0.0
  0.0     1.0      0.0      0.0         0.0      0.0      0.0      0.0
  ⋮                                 ⋱  ⋮
  0.0     0.0      0.0      0.0         0.0      0.0      1.0      0.0
  0.0     0.0      0.0      0.0      …  0.0      0.0      0.0      1.0
  0.0     0.0      0.0      0.0         1.0      0.0      0.0      0.0
  0.0     0.0      0.0      0.0         0.0      1.0      0.0      0.0
  0.0     0.0      0.0      0.0         0.0      0.0      1.0      0.0
  0.0     0.0      0.0      0.0         0.0      0.0      0.0      1.0
  0.0     0.0      0.0      0.0      …  5.81673  1.0      1.0      0.0
  0.0     0.0      0.0      0.0         1.0      14.9075  0.0      1.0
  0.0     0.0      0.0      0.0         1.0      0.0      11.6526  1.0
  0.0     0.0      0.0      0.0         0.0      1.0      1.0      21.607
```

- **Problem:** Inefficient!
- Even for a full coupling matrix, the Hamiltonian is very sparse!

Most elements are zero!

- Elements in matrix: 2^{2N}
- Non-zero elements: $\mathcal{O}(2^N)$

This example:

```
julia> sum((H .> 1e-16))
10698
```

```
julia> sum((H .> 1e-16)) / 2^20
0.010202407836914062
```

Only 1% filled … let's use **Sparse Matrices!**

Lecture 2 - Sparse Matrices!

- **Minimal modifications**

```
# Construct lowering operators
using SparseArrays
N = 10
sm = sparse([0 1; 0 0])
sms = Vector{}(undef, N)

for ii = 1:N
    ldim = 2^(ii-1)
    rdim = 2^(N - ii)
    sms[ii] = kron(sparse(I, ldim, ldim), sm, sparse(I, rdim, rdim))
end
```

```
# Construct Hamiltonian
using SparseArrays
Jij = rand(N,N) # random couplings
hx = 1;

H = spzeros(2^N, 2^N)
for ii = 1:N
    H += hx .* (sms[ii]' + sms[ii]) # sx
    szii = (sms[ii]'*sms[ii] - sms[ii]*sms[ii]') # sz spin ii
    for jj = (ii+1):N
        szjj = (sms[jj]'*sms[jj] - sms[jj]*sms[jj]') # sz spin jj
        H += Jij[ii,jj] .* szii * szjj
    end
end
```

Full vs. sparse version

```
61.911068 seconds
 0.004216 seconds
```

Lecture 2 - Hilbert space construction in a many-body system

- **Remark:** Initial product states can also easily produced with Kronecker products

$$|\psi_0\rangle = |\downarrow\uparrow\downarrow\uparrow\rangle = |\downarrow\rangle \otimes |\uparrow\rangle \otimes |\downarrow\rangle \otimes |\uparrow\rangle \quad \Leftrightarrow$$

```
spd = [1; 0]
spu = [0; 1]
psi0 = kron(spd,spu,spd,spu)
```

…or even more simply: $|\psi_0\rangle = |\downarrow\downarrow\downarrow\downarrow\rangle \quad \Leftrightarrow$

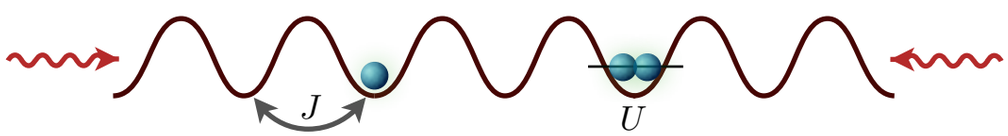
```
psi0 = zeros(2^N)
psi0[1] = 1.0
```

- **Remark:** We want to compute expectation values, this can e.g. again be easily achieved using our sparse operators

$$\langle\psi(t)|\,\hat{\sigma}_i^z\,|\psi(t)\rangle \quad \Leftrightarrow$$

```
szjj = (sms[jj]'*sms[jj] - sms[jj]*sms[jj]') # sz spin jj
expc_szjj = real(psi'*szjj*psi)
```

- **Remark:** For a system of bosons in a lattice, very similar construction possible



$$\hat{H} = -J\sum_i(\hat{b}_i\hat{b}_{i+1}^\dagger + \hat{b}_i^\dagger\hat{b}_{i+1}) + \frac{U}{2}\sum_i \hat{b}_i^\dagger\hat{b}_i^\dagger\hat{b}_i\hat{b}_i$$

But: *Number not conserved!*

$$D = n_{\max}^N$$

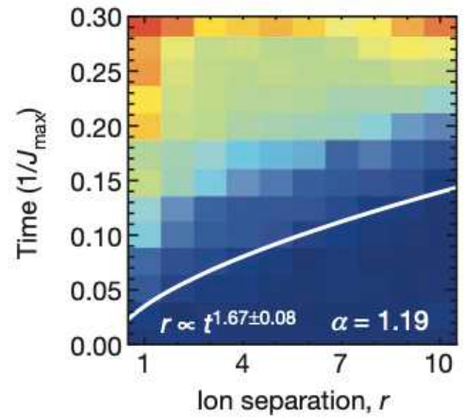
Introduce cutoff of bosons per site

Define bosonic field operators and use Kroneckers as before

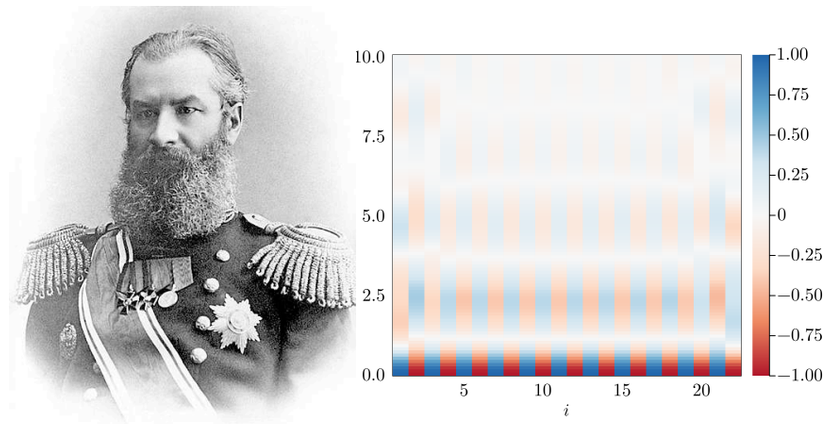
```
# |n=0> = [1;0;0;0; ...]
# |n=2> = [0;0;1;0; ...]
# b |n=2> = sqrt(n) * [0;1;0;0; ...]
nm = 5 # cutoff for max. nm-1 bosons)
b = sparse(1:(nm-1), 2:nm, sqrt.(1:(nm-1)), nm, nm)
```

Plan for today

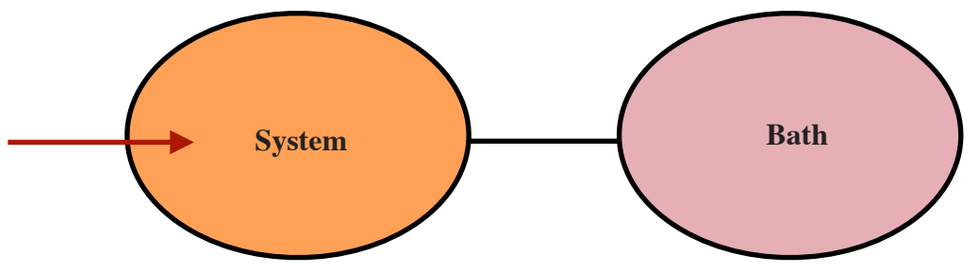
- **Part 2.1:** The many-body problem of the day: Long-range spin model dynamics. How to numerically construct Hamiltonians.



- **Part 2.2:** Simulation of dynamics with sparse Hamiltonians: Krylov space methods. Applications to spin-model dynamics.



- **Part 2.3:** Methods for open system dynamics: Density matrix linearization and quantum trajectories.



Krylov space methods

- The goal is to compute dynamics of some initial state

$$\frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle \qquad |\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\hat{H}t}\,|\psi_0\rangle \qquad \ldots \textit{with a generally very sparse matrix}$$

- We could just use Runge-Kutta for this, but it can be overkill:

i) The problem is linear
ii) The Hamiltonian is time-independent

- A more efficient method can be to compute the matrix exponential directly!

- **Problem:** A full matrix exponentiation turns the sparse Hamiltonian into a full matrix

E.g.: Compute matrix exponential by full diagonalization

$$\hat{H}\,|E_n\rangle = E_n\,|E_n\rangle \qquad \hat{H} = \sum_n E_n\,|E_n\rangle\langle E_n| \qquad \mathrm{e}^{-\mathrm{i}\hat{H}t} = \sum_{k=0}\frac{(-\mathrm{i}t\hat{H})^k}{k!} = \sum_n \mathrm{e}^{-\mathrm{i}tE_n}\,|E_n\rangle\langle E_n|$$

sparse ~D elements — *D energy eigenstates* — *each ~D elements* — *in general: Full DxD matrix :(*

- **On the other hand:** We're only interested in the state after applying the matrix exponential

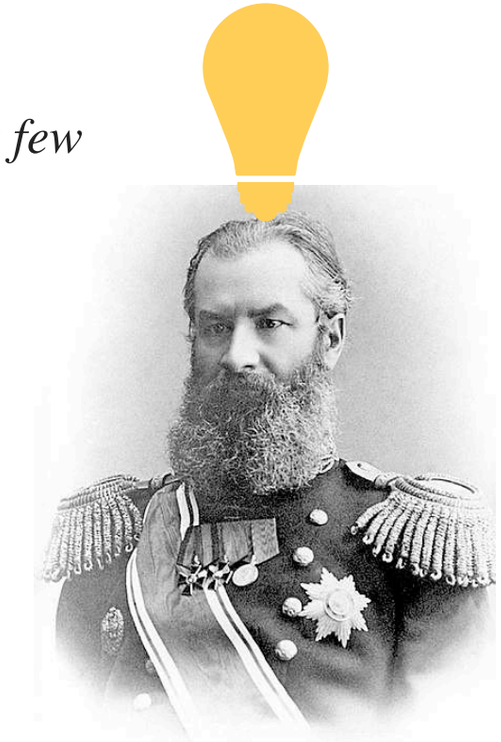
$$\mathrm{e}^{-\mathrm{i}\hat{H}t}\,|\psi_0\rangle = \sum_{k=0}\frac{(-\mathrm{i}t\hat{H})^k}{k!}\,|\psi_0\rangle \equiv \sum_{k=0}\frac{\hat{A}^k}{k!}\,|\psi_0\rangle$$

Maybe we only need a few terms in this sum?

Krylov space methods

$$e^{-i\hat{H}t}\,|\psi_0\rangle = \sum_{k=0} \frac{(-it\hat{H})^k}{k!}\,|\psi_0\rangle \equiv \sum_{k=0} \frac{\hat{A}^k}{k!}\,|\psi_0\rangle$$

Maybe we only need a few terms in this sum?



Aleksey Nikolaevich Krylov (1863 - 1945)

- Starting from the initial state, a good idea seems to be using the following space:
- **Krylov space:** $\text{span}\left(\hat{A}^0\,|\psi_0\rangle\,, \hat{A}^1\,|\psi_0\rangle\,, \hat{A}^2\,|\psi_0\rangle\,, \ldots, \hat{A}^{m-1}\,|\psi_0\rangle\right)$

(here formulated in bra/ket notation, but concept is general)

Krylov wrote about 300 papers and books. They span a wide range of topics, including shipbuilding, magnetism, artillery, mathematics, astronomy, and geodesy.

- **Problem:** The states are not orthogonal (in fact with increasing *k* they become more and more parallel)

Example, let's for a sec take the matrix to be a hermitian Hamiltonian $\hat{A} = \hat{H}$

$$|\psi_0\rangle = \sum_n c_n\,|E_n\rangle \qquad \hat{H}\,|E_n\rangle = E_n\,|E_n\rangle \qquad E_1 > E_2 > E_3 > \ldots$$

$$|\psi_k\rangle \equiv \hat{H}^k\,|\psi_0\rangle = \sum_n c_n E_n^k\,|E_n\rangle$$

- When renormalizing: $\dfrac{|\psi_k\rangle}{\|\,|\psi_k\rangle\,\|} \approx |E_1\rangle \qquad k \gg 1$

Repeated application gives the highest energy state!
*Known as **power method***

$c_1 \neq 0$

Krylov space methods - Arnoldi iteration

- **Krylov space:** $\text{span}\left(\hat{A}^0 |\psi_0\rangle, \hat{A}^1 |\psi_0\rangle, \hat{A}^2 |\psi_0\rangle, \ldots, \hat{A}^{m-1} |\psi_0\rangle\right)$

  (here formulated in bra/ket notation, but concept is general)

- **Problem:** The states are not orthogonal (in fact with increasing *k* they become more and more parallel)

- **Arnoldi iteration:** Create an orthonormal basis from the Krylov vectors (using improved Gram-Schmidt)

$|\psi_0\rangle = |\psi_0\rangle / \| |\psi_0\rangle \|$ *(normalization just in case)*

1st iteration

$|\psi_1\rangle = \hat{A} |\psi_0\rangle$

$h_{1,1} \equiv \langle\psi_0|\psi_1\rangle \quad |\psi_1\rangle = |\psi_1\rangle - h_{1,1} |\psi_0\rangle$

$h_{2,1} \equiv \| |\psi_1\rangle \| \quad |\psi_1\rangle = |\psi_1\rangle / h_{2,1}$

ith - iteration

$|\psi_i\rangle = \hat{A} |\psi_{i-1}\rangle$

$h_{1,i} \equiv \langle\psi_0|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{1,i} |\psi_0\rangle$

$h_{2,i} \equiv \langle\psi_1|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{2,i} |\psi_1\rangle$

$\ldots$

$h_{i,i} \equiv \langle\psi_{i-1}|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{i+1,i} |\psi_{i-1}\rangle$

$h_{i+1,1} \equiv \| |\psi_i\rangle \| \quad |\psi_i\rangle = |\psi_i\rangle / h_{i+1,1}$

Walter Edwin Arnoldi
(1917 - 1995)

Krylov space methods - Arnoldi iteration

- **Krylov space:** $\text{span}\left(\hat{A}^0|\psi_0\rangle, \hat{A}^1|\psi_0\rangle, \hat{A}^2|\psi_0\rangle, \ldots, \hat{A}^{m-1}|\psi_0\rangle\right)$

$$|\psi_i\rangle = \hat{A}|\psi_{i-1}\rangle \qquad \textit{ith - iteration}$$

$$h_{1,i} \equiv \langle\psi_0|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{1,i}|\psi_0\rangle$$

$$h_{2,i} \equiv \langle\psi_1|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{2,i}|\psi_1\rangle$$

$$\ldots$$

$$h_{i,i} \equiv \langle\psi_{i-1}|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{i+1,i}|\psi_{i-1}\rangle$$

$$h_{i+1,1} \equiv \| |\psi_i\rangle \| \quad |\psi_i\rangle = |\psi_i\rangle / h_{i+1,1}$$

after i iterations, we have i+1 orthogonal vectors, i.e. we have the "semi-unitary" matrix

$$\mathbf{Q}_i = \begin{pmatrix} |\psi_0\rangle & |\psi_1\rangle & \ldots & |\psi_i\rangle \end{pmatrix}$$

$$D \times (i+1) \qquad \mathbf{Q}_i\mathbf{Q}_i^\dagger = \mathbb{1}$$

- $\mathbf{Q}_i$ is a projection matrix to the Krylov basis

- What is the matrix **h?**

$$\mathbf{h}_i = \begin{pmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,i} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,i} \\ 0 & h_{3,2} & \ldots & h_{3,i} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & h_{i+1,i} \end{pmatrix}$$

$$(i+1) \times i$$

Upper Hessenberg form

The iteration can be written in the matrix form:

$$\hat{A}\mathbf{Q}_i = \mathbf{Q}_{i+1}\mathbf{h}_i \qquad \textit{(exercise: show)}$$

Example: i=1

$$|\psi_1\rangle = \left(\hat{A}|\psi_0\rangle - h_{1,1}|\psi_0\rangle\right)/h_{2,1}$$

$$\hat{A}|\psi_0\rangle = h_{1,1}|\psi_0\rangle + h_{2,1}|\psi_1\rangle$$

$$\hat{A}(|\psi_0\rangle) = \begin{pmatrix} |\psi_0\rangle & |\psi_1\rangle \end{pmatrix}\begin{pmatrix} h_{1,1} \\ h_{2,1} \end{pmatrix}$$

Krylov space methods - Arnoldi iteration

- **Krylov space:** $\text{span}\left(\hat{A}^0|\psi_0\rangle, \hat{A}^1|\psi_0\rangle, \hat{A}^2|\psi_0\rangle, \ldots, \hat{A}^{m-1}|\psi_0\rangle\right)$

$$|\psi_i\rangle = \hat{A}|\psi_{i-1}\rangle \qquad \textit{ith - iteration}$$

$$h_{1,i} \equiv \langle\psi_0|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{1,i}|\psi_0\rangle$$

$$h_{2,i} \equiv \langle\psi_1|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{2,i}|\psi_1\rangle$$

$$\ldots$$

$$h_{i,i} \equiv \langle\psi_{i-1}|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{i+1,i}|\psi_{i-1}\rangle$$

$$h_{i+1,1} \equiv \| |\psi_i\rangle \| \quad |\psi_i\rangle = |\psi_i\rangle / h_{i+1,1}$$

after i iterations, we have i+1 orthogonal vectors, i.e. we have the "semi-unitary" matrix

$$\mathbf{Q}_i = \begin{pmatrix} |\psi_0\rangle & |\psi_1\rangle & \ldots & |\psi_i\rangle \end{pmatrix}$$

$$D \times (i+1) \qquad \mathbf{Q}_i\mathbf{Q}_i^\dagger = \mathbb{1}$$

- What is the matrix **h?**

The iteration can be written in the matrix form:

$$\hat{A}\mathbf{Q}_i = \mathbf{Q}_{i+1}\mathbf{h}_i$$

$$\mathbf{Q}_{i+1} = \begin{pmatrix} |\psi_0\rangle & |\psi_1\rangle & \ldots & |\psi_i\rangle & \cancel{|\psi_{i+1}\rangle} \end{pmatrix} \qquad \tilde{\mathbf{h}}_i = \begin{pmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,i} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,i} \\ 0 & h_{3,2} & \ldots & h_{3,i} \\ \vdots & \vdots & \ddots & \vdots \\ \cancel{0} & \cancel{0} & \ldots & h_{i+1,i} \end{pmatrix}$$

- Terminating means:

$$\mathbf{Q}_{i+1}\mathbf{h}_i \approx \mathbf{Q}_i\tilde{\mathbf{h}}_i$$

- Then: $\hat{A}\mathbf{Q}_i \approx \mathbf{Q}_i\tilde{\mathbf{h}}_i$ $\qquad \tilde{\mathbf{h}}_i \approx \mathbf{Q}_i^\dagger\hat{A}\mathbf{Q}_i \qquad \hat{A} \approx \mathbf{Q}_i\tilde{\mathbf{h}}_i\mathbf{Q}_i^\dagger$

- The matrix **h** is an approximation of the matrix **A** in (the much smaller) Krylov space!

Krylov space methods - Arnoldi iteration

- **Krylov space:** $\text{span}\left(\hat{A}^0|\psi_0\rangle, \hat{A}^1|\psi_0\rangle, \hat{A}^2|\psi_0\rangle, \ldots, \hat{A}^{m-1}|\psi_0\rangle\right)$

$$|\psi_i\rangle = \hat{A}|\psi_{i-1}\rangle \qquad \textit{ith - iteration}$$

$$h_{1,i} \equiv \langle\psi_0|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{1,i}|\psi_0\rangle$$

$$h_{2,i} \equiv \langle\psi_1|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{2,i}|\psi_1\rangle$$

$$\ldots$$

$$h_{i,i} \equiv \langle\psi_{i-1}|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{i+1,i}|\psi_{i-1}\rangle$$

$$h_{i+1,1} \equiv \| |\psi_i\rangle \| \quad |\psi_i\rangle = |\psi_i\rangle / h_{i+1,1}$$

after i iterations, we have i+1 orthogonal vectors, i.e. we have the "semi-unitary" matrix

$$\mathbf{Q}_i = \begin{pmatrix} |\psi_0\rangle & |\psi_1\rangle & \ldots & |\psi_i\rangle \end{pmatrix}$$

$$D \times (i+1) \qquad \mathbf{Q}_i \mathbf{Q}_i^\dagger = \mathbb{1}$$

- The matrix **h** is an approximation of the matrix **A** in (the much smaller) Krylov space!

- The idea is to now do exact diagonalization on this matrix and then to transform it back to the full space

- **Remark:** For hermitian matrices, e.g. $\hat{A} = \hat{H}$ the matrix **h** is even only tri-diagonal. Then the iteration to obtain **h** is even simpler, it's called **Lanczos** iteration. It's very useful to compute a few eigenvalues and eigenstates of very large sparse Hamiltonians.

$$\mathbf{h}_i = \begin{pmatrix} h_{1,1} & h_{1,2} & 0 & 0 & \ldots \\ h_{2,1} & h_{2,2} & h_{2,3} & 0 & \ldots \\ 0 & h_{3,2} & h_{3,2} & h_{3,2} & \ldots \\ 0 & 0 & h_{4,3} & h_{4,3} & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Matrix exponential on Krylov space

$$|\psi_i\rangle = \hat{A}\,|\psi_{i-1}\rangle$$

$$h_{1,i} \equiv \langle\psi_0|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{1,i}\,|\psi_0\rangle$$

$$h_{2,i} \equiv \langle\psi_1|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{2,i}\,|\psi_1\rangle$$

$$\cdots$$

$$h_{i,i} \equiv \langle\psi_{i-1}|\psi_i\rangle \quad |\psi_i\rangle = |\psi_i\rangle - h_{i+1,i}\,|\psi_{i-1}\rangle$$

$$h_{i+1,1} \equiv \| \, |\psi_i\rangle \, \| \quad |\psi_i\rangle = |\psi_i\rangle \, / h_{i+1,1}$$

- For the matrix exponential applied to $|\psi_0\rangle$

… approximation is formally

$$\mathrm{e}^{\hat{A}}\,|\psi_0\rangle \approx \mathbf{Q}_m \mathrm{e}^{\mathbf{h}} \mathbf{Q}_m^\dagger\,|\psi_0\rangle$$

… but since $|\psi_0\rangle$ is the first row in $\mathbf{Q}_m^\dagger$, and all other rows are orthogonal, we just need to take the first column of $\mathrm{e}^{\mathbf{h}}$

```
# Compute approximation to exp(A)*psi0
# ... with m Krylov basis states
function arnoldi_exp(A, psi0, m)

    T = typeof(A[1]) # real or complex
    D = length(psi0)

    Q = zeros(T, D, m) # the projection matrix
    h = zeros(T, m, m) # Krylov projection of A

    Q[:,1] = psi0   # assumed normalized
    for ii = 1:(m-1)
        psi_i = A*Q[:,ii]
        for jj = 1:ii
            h[jj,ii] = Q[:,jj]' * psi_i
            psi_i -= h[jj,ii] .* Q[:,jj]
        end
        h[ii+1, ii] = norm(psi_i)
        Q[:,ii+1] = psi_i ./ h[ii+1, ii]
    end

    # now return the matrix exponential
    return Q * exp(h)[:,1]

end
```

Matrix exponential on Krylov space

- Test with full random Hamiltonian $\hat{H}$

$$\hat{A} = -\mathrm{i}\Delta t \hat{H}$$

```
D = 1000; m = 20
H = rand(ComplexF64, D,D); H += H'
A = -1im .* 0.1 .* H

psi0 = rand(ComplexF64, D); psi0 ./= norm(psi0)

@time psi_ed = exp(A)*psi0
@time psi_krylov = arnoldi_exp(A, psi0, m)
@show abs(psi_ed' * psi_krylov)^2
```

```
  0.676812 seconds (16 allocations: 91.584 MiB)
  0.007632 seconds (831 allocations: 12.931 MiB)
abs(psi_ed' * psi_krylov) ^ 2 = 0.9999999999999936
```

Krylov approximation is already exact up to machine precision with m ~ 20. Speed-up of two orders of magnitude. For sparse matrices even more!

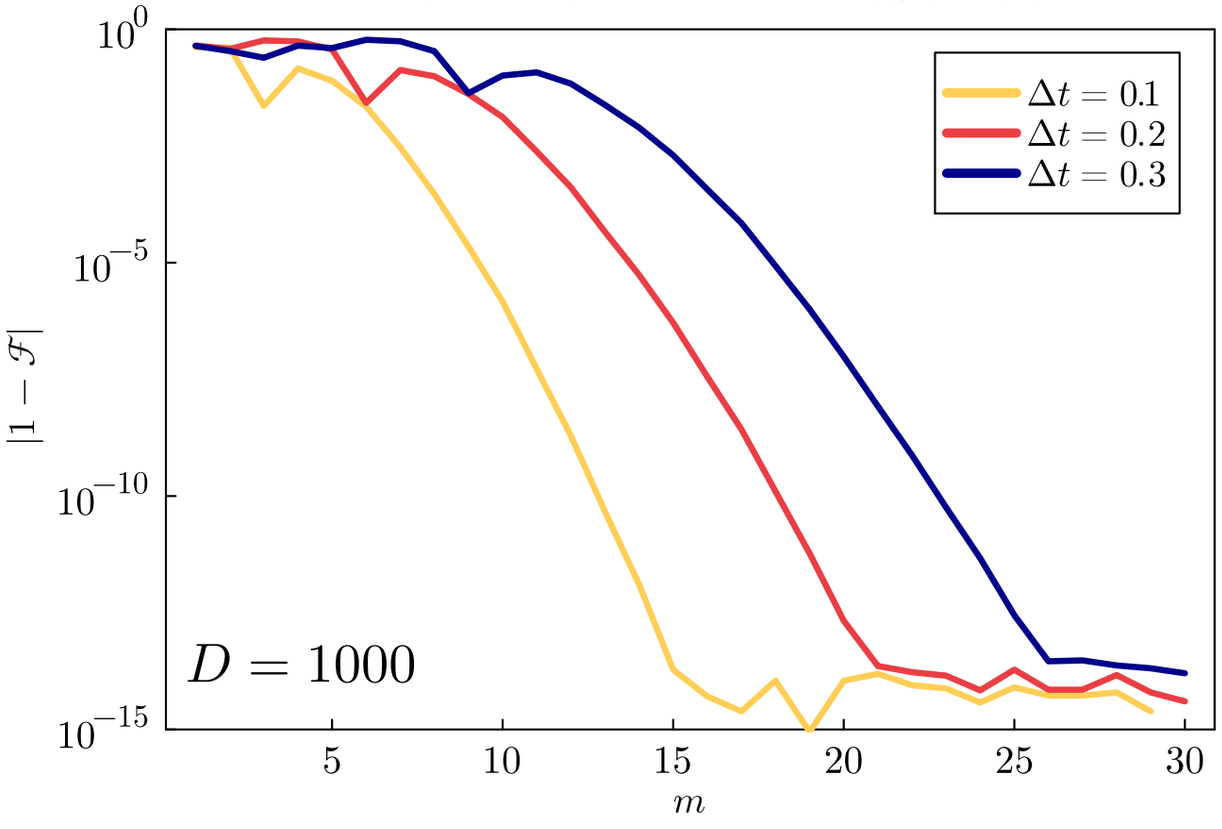
```
# Compute approximation to exp(A)*psi0
# ... with m Krylov basis states
function arnoldi_exp(A, psi0, m)

    T = typeof(A[1]) # real or complex
    D = length(psi0)

    Q = zeros(T, D, m) # the projection matrix
    h = zeros(T, m, m) # Krylov projection of A

    Q[:,1] = psi0   # assumed normalized
    for ii = 1:(m-1)
        psi_i = A*Q[:,ii]
        for jj = 1:ii
            h[jj,ii] = Q[:,jj]' * psi_i
            psi_i -= h[jj,ii] .* Q[:,jj]
        end
        h[ii+1, ii] = norm(psi_i)
        Q[:,ii+1] = psi_i ./ h[ii+1, ii]
    end

    # now return the matrix exponential
    return Q * exp(h)[:,1]

end
```

Matrix exponential on Krylov space

- Test with full random Hermitian matrix $\hat{H}$

$$\hat{A} = -\mathrm{i}\Delta t \hat{H}$$

$$\mathcal{F} = |\langle\psi_0| (\mathrm{e}^{\hat{A}})^\dagger \mathbf{Q}_m \mathrm{e}^{\mathbf{h}} \mathbf{Q}_m^\dagger |\psi_0\rangle|^2$$



A larger "time-step" will need a larger Krylov space dimension.

```
# Compute approximation to exp(A)*psi0
# ... with m Krylov basis states
function arnoldi_exp(A, psi0, m)

    T = typeof(A[1]) # real or complex
    D = length(psi0)

    Q = zeros(T, D, m) # the projection matrix
    h = zeros(T, m, m) # Krylov projection of A

    Q[:,1] = psi0   # assumed normalized
    for ii = 1:(m-1)
        psi_i = A*Q[:,ii]
        for jj = 1:ii
            h[jj,ii] = Q[:,jj]' * psi_i
            psi_i -= h[jj,ii] .* Q[:,jj]
        end
        h[ii+1, ii] = norm(psi_i)
        Q[:,ii+1] = psi_i ./ h[ii+1, ii]
    end

    # now return the matrix exponential
    return Q * exp(h)[:,1]

end
```

Matrix exponential on Krylov space

- **Remark:** The implementation on the right is very simple, we did not implement any stopping condition and didn't discuss any (smart) error estimation

- There are many implementations out there, e.g. *KrylovKit* for Julia (or *ExpoKit*), etc.

```julia
using KrylovKit

psi, info = exponentiate(A, 1.0, psi0;
                         krylovdim=m, tol=1e-12)
```

```julia
# Compute approximation to exp(A)*psi0
# ... with m Krylov basis states
function arnoldi_exp(A, psi0, m)

    T = typeof(A[1]) # real or complex
    D = length(psi0)

    Q = zeros(T, D, m) # the projection matrix
    h = zeros(T, m, m) # Krylov projection of A

    Q[:,1] = psi0   # assumed normalized
    for ii = 1:(m-1)
        psi_i = A*Q[:,ii]
        for jj = 1:ii
            h[jj,ii] = Q[:,jj]' * psi_i
            psi_i -= h[jj,ii] .* Q[:,jj]
        end
        h[ii+1, ii] = norm(psi_i)
        Q[:,ii+1] = psi_i ./ h[ii+1, ii]
    end

    # now return the matrix exponential
    return Q * exp(h)[:,1]

end
```

Applying Krylov space evolution

- Long-range transverse Ising model: $\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j} \hat{\sigma}_i^z \hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha}$

```
Jij = zeros(N,N)
alpha = 3
for ii = 1:N
    for jj = (ii+1):N
        Jij[ii,jj] = 1/(abs(ii-jj)^alpha)
    end
end
```

… sparse, as discussed earlier

```
H = spzeros(2^N, 2^N)
for ii = 1:N
    H += hx .* (sms[ii]' + sms[ii]) # sx
    szii = (sms[ii]'*sms[ii] - sms[ii]*sms[ii]') # sz spin ii
    for jj = (ii+1):N
        szjj = (sms[jj]'*sms[jj] - sms[jj]*sms[jj]') # sz spin jj
        H += Jij[ii,jj] .* szii * szjj
    end
end
```

- Initial state (highly excited, not an eigenstate)

Néel state

$$|\psi_0\rangle = |\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow \ldots\rangle$$

```
psi = zeros(2^N)
psi[1] = 1.0
for oo = 1:2:N
    psi = sms[oo]' * psi
end
```

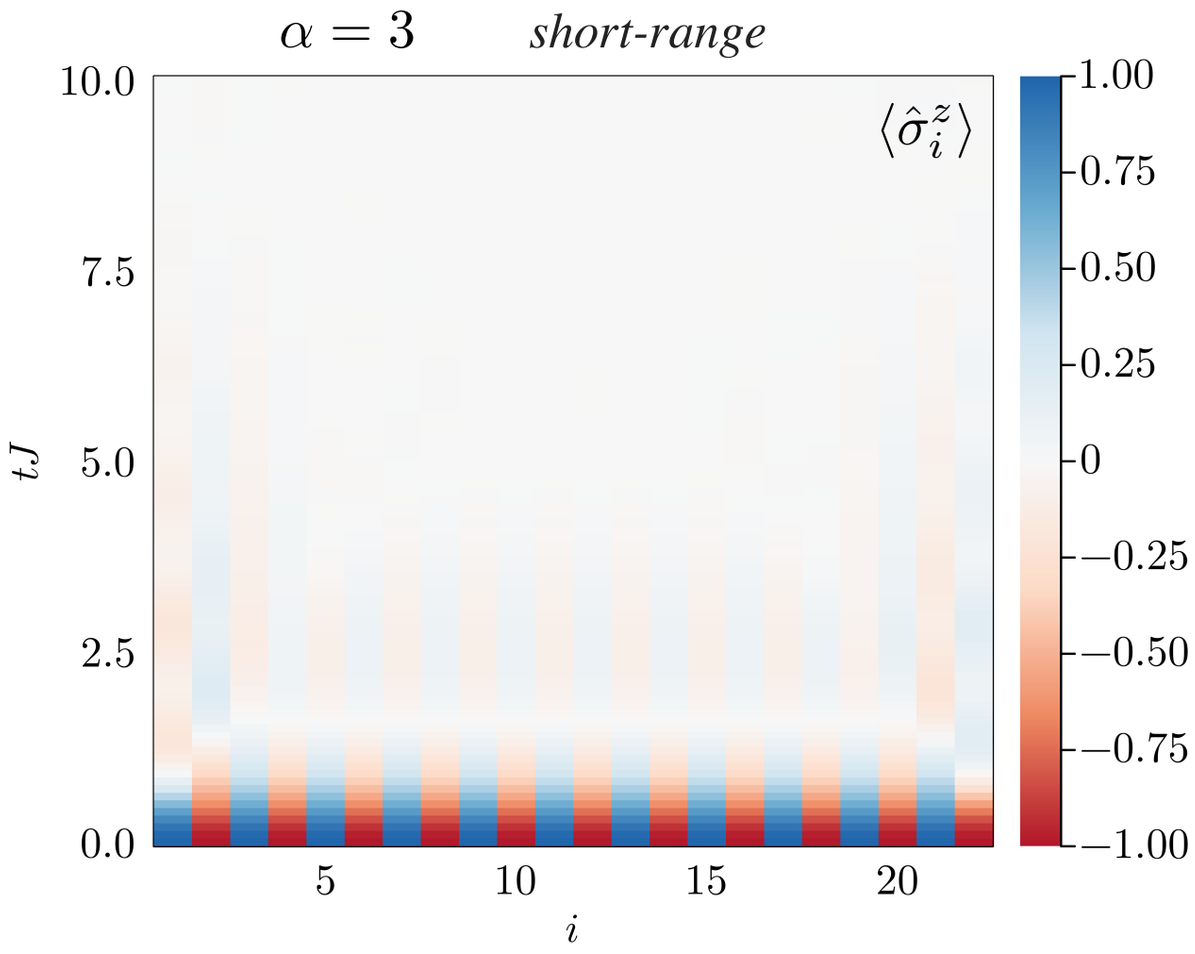
- Then, applying Krylov time-evolution steps

```
psi, info = exponentiate(H, -1im * dt, psi)
```

Spin-model evolution with Krylov space matrix exponential

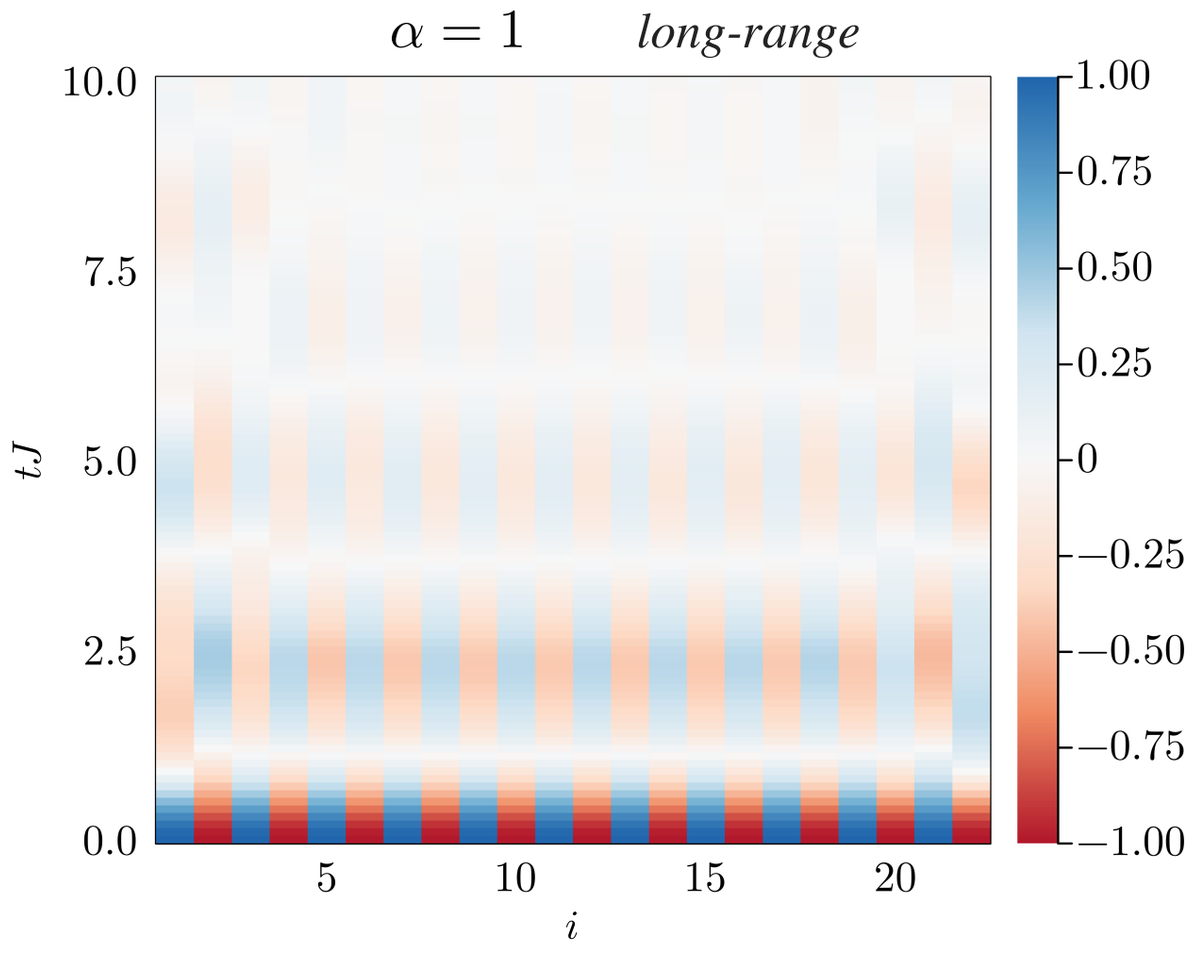
- Evolution of local spin-z component
- $N = 22$ spins

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad h_x = J$$



For the short range case, correlation build-up leads to local mixed states. For example, for sufficiently long times, the reduced density matrices will become fully mixed.

$$\hat{\rho}_i = \mathrm{tr}_{j\neq i}(|\psi(t)\rangle\langle\psi(t)|) \approx \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} \qquad \mathrm{tr}(\hat{\rho}_i\hat{\sigma}_i^z) \approx 0 \qquad t \gg J^{-1}$$



In the long-range case, dynamics is more collective

$$\alpha = 0$$

$$\hat{H}_{\mathrm{TI}} = J\sum_{i<j} \hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x$$

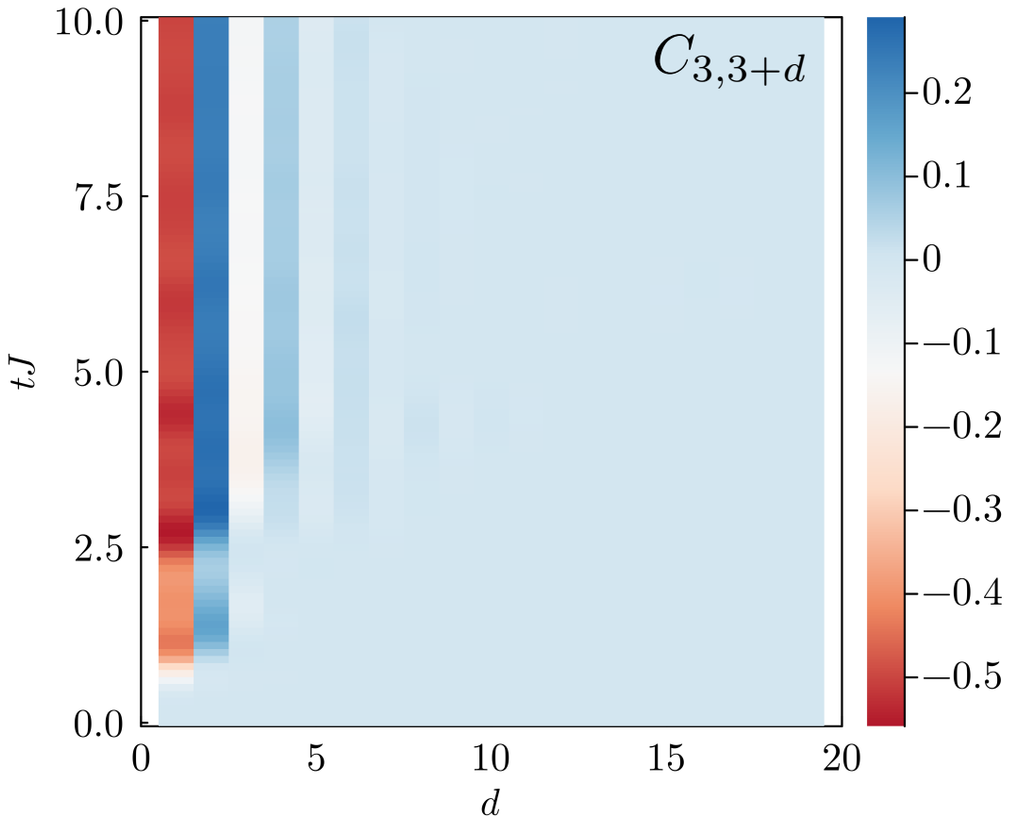
Spin-model evolution with Krylov space matrix exponential

- Evolution of two-spin correlations
- $N = 22$ spins

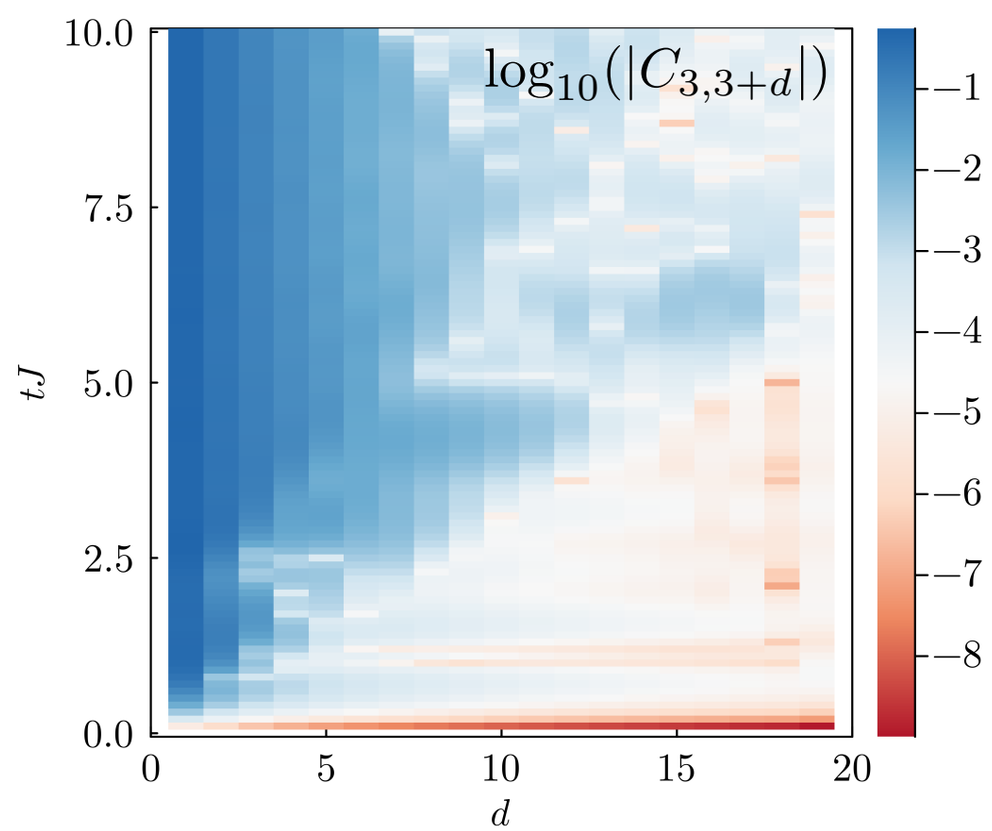
$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j} \hat{\sigma}_i^z \hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha}$$

$$C_{i,j} = \langle \hat{\sigma}_i^z \hat{\sigma}_j^z \rangle - \langle \hat{\sigma}_i^z \rangle \langle \hat{\sigma}_j^z \rangle$$

$\alpha = 3$ *short-range*



"connected correlation", subtract classical part



There is a spreading out in a "light-cone". Even the two-spin correlations become very small over long-distances.

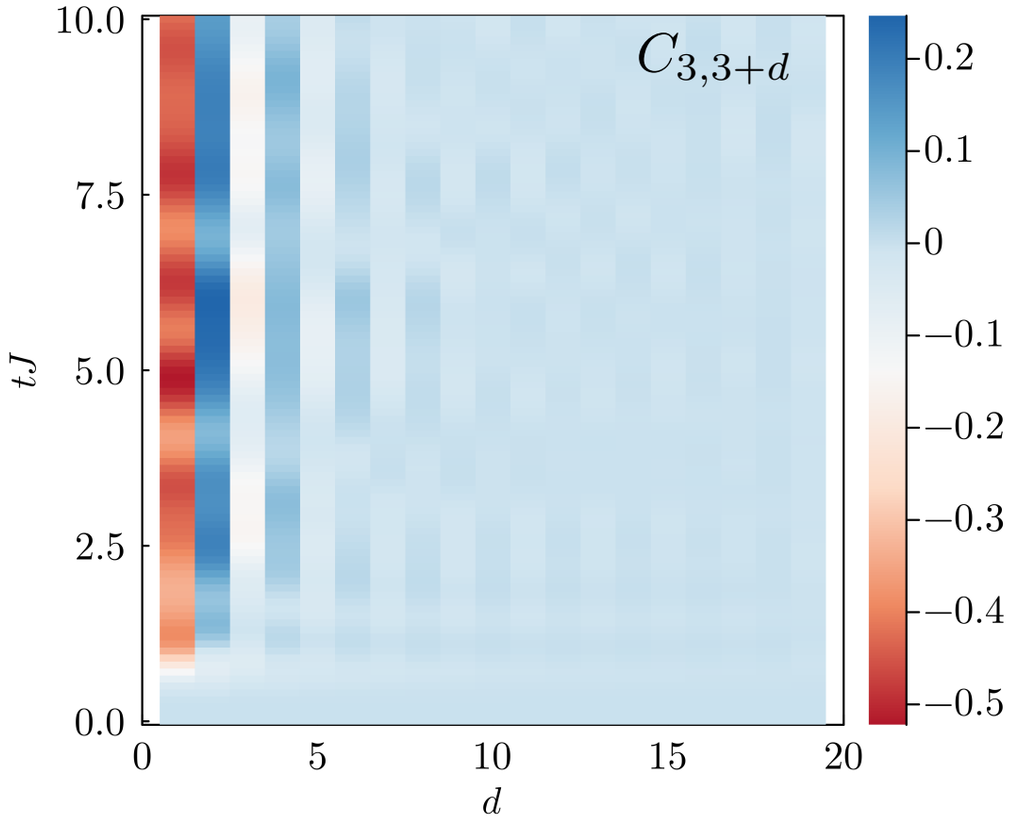
Spin-model evolution with Krylov space matrix exponential

- Evolution of two-spin correlations
- $N = 22$ spins

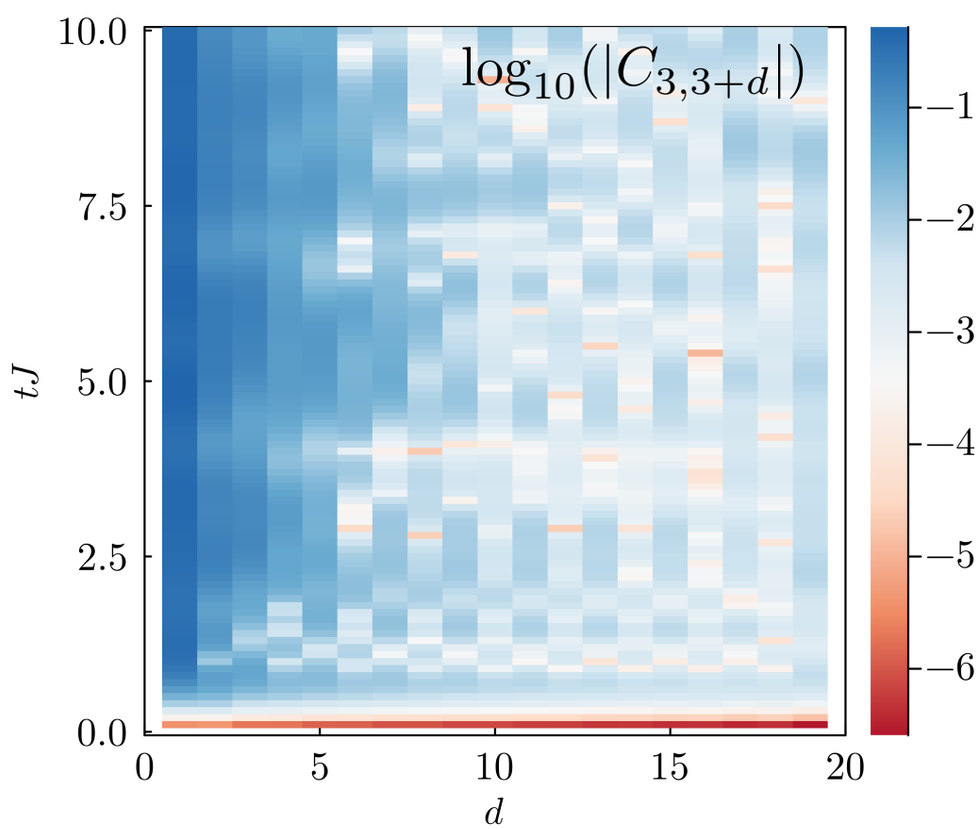
$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha}$$

$$C_{i,j} = \langle\hat{\sigma}_i^z\hat{\sigma}_j^z\rangle - \langle\hat{\sigma}_i^z\rangle\langle\hat{\sigma}_j^z\rangle$$

$\alpha = 1$ *long-range*



"connected correlation", subtract classical part



For long-range couplings the light-cone disappears

- Complex information spreading physics as competition between long and short-range terms … further reading:

P. Calabrese and J. Cardy, J. Stat. Mech, P04010 (2005)

J. Eisert, M. van den Worm, S. R. Manmana, M. Kastner, Phys. Rev. Lett. 111, 260401 (2013)

J. Schachenmayer, B. P. Lanyon, C. F. Roos, A. J. Daley, Phys. Rev. X 3, 031015 (2013)

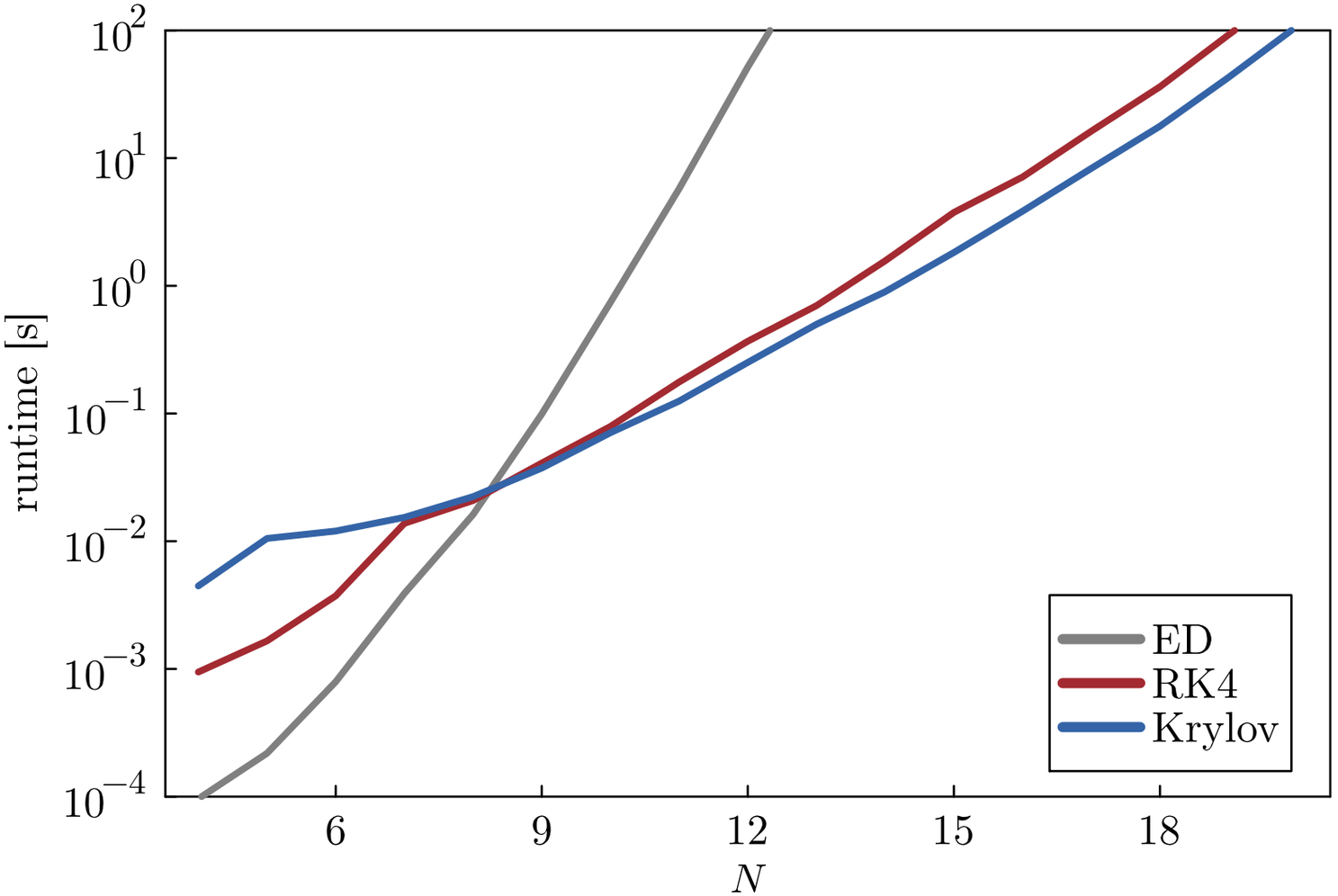
Z.-X. Gong, M. Foss-Feig, Fernando G. S. L. Brandão, and A. V. Gorshkov, Phys. Rev. Lett. 119 050501 (2017)

L. Cevolani, J. Despres, G. Carleo, L. Tagliacozzo, L. Sanchez-Palencia, Phys. Rev. B 98, 024302 (2018)

Methods - Comparison

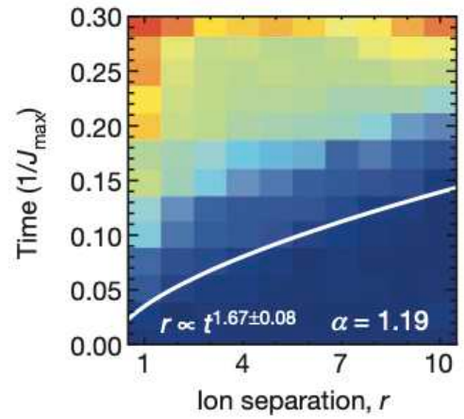
- Evolution of two-spin correlations $\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x \sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad \alpha = 1$
- Simulation up to $tJ \leq 5$ *… on Apple M2 Macbook (on battery)* $|\psi_0\rangle = |\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\ldots\rangle$
- For fair comparison:

ED	*(no time-step)*	$\lvert 1 - \lVert \lvert\psi(t)\rangle \rVert \rvert < 10^{-14}$
RK4	$\Delta tJ = 0.005$	$\Leftrightarrow \lvert 1 - \lVert \lvert\psi(t)\rangle \rVert \rvert < 10^{-5}$
Krylov	$\Delta tJ = 0.1$	$\Leftrightarrow \lvert 1 - \lVert \lvert\psi(t)\rangle \rVert \rvert < 10^{-13}$

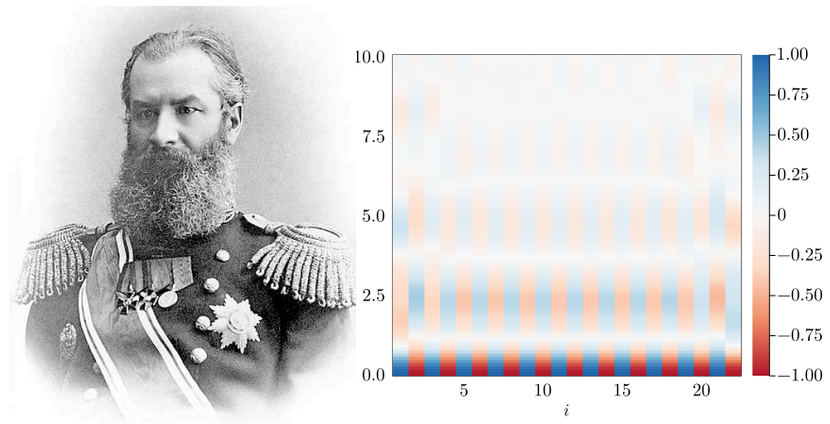


Plan for today

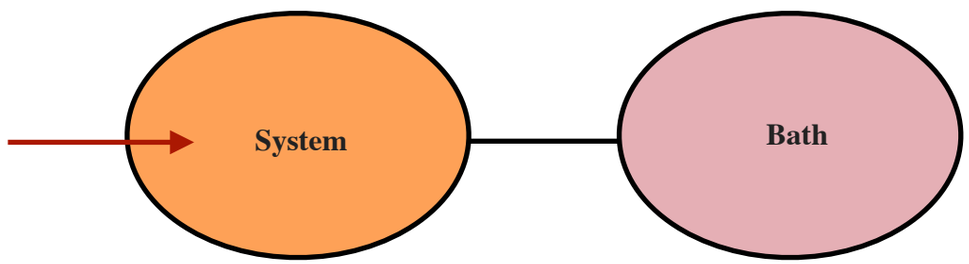
- **Part 2.1:** The many-body problem of the day: Long-range spin model dynamics. How to numerically construct Hamiltonians.



- **Part 2.2:** Simulation of dynamics with sparse Hamiltonians: Krylov space methods. Applications to spin-model dynamics.



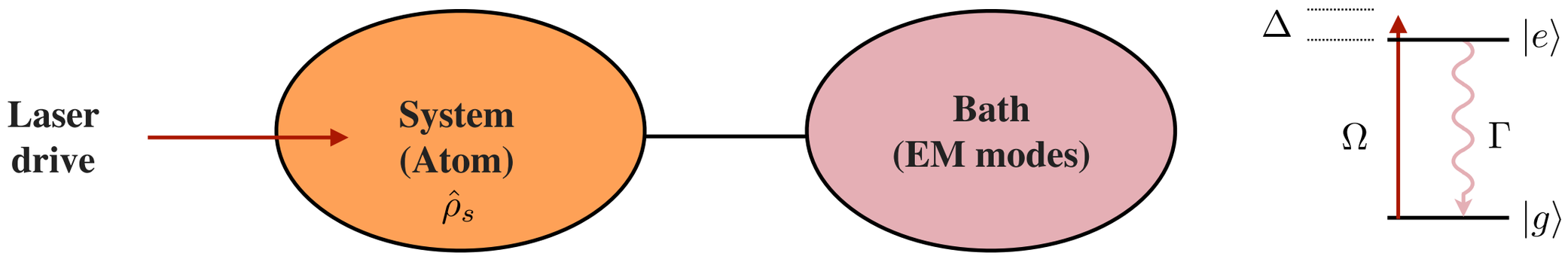
- **Part 2.3:** Methods for open system dynamics: Density matrix linearization and quantum trajectories.



Lecture 2 - Lindblad master equations - Review

- Here we will focus on Lindblad master equations: We have a **System** and a **large bath:**

Example: Consider an atom coupled to the electromagnetic vacuum modes



- Effectively a Lindblad master equation describes the dynamics of the system only (the bath is traced out), the system needs to be described by a density matrix $\hat{\rho}_s$

- Several approximations on the way:

For the atomic decay problem: Dipole approximation, Rotating wave approximation, ***Markov approximation****, …*

Open quantum system lecture

$$\frac{d}{dt}\hat{\rho}_S = \mathcal{L}(\hat{\rho}_S) = -\mathrm{i}\left[\hat{H}_S, \hat{\rho}\right] + \sum_\eta \left(2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger - \hat{L}_\eta^\dagger \hat{L}_\eta \hat{\rho}_S - \hat{\rho}_S \hat{L}_\eta^\dagger \hat{L}_\eta\right)$$

Lindblad jump operators

- E.g. for a single decaying atom *"Optical Bloch equations"* $\hat{L} = \sqrt{\frac{\Gamma}{2}}\hat{\sigma}^-$

$$\frac{d}{dt}\hat{\rho}_S = -\mathrm{i}\left[\hat{H}_{0A}, \hat{\rho}_S\right] + \frac{\Gamma}{2}\left(-\hat{\sigma}^+\hat{\sigma}^-\hat{\rho}_S - \hat{\rho}_S\hat{\sigma}^+\hat{\sigma}^- + 2\hat{\sigma}^-\hat{\rho}_S\hat{\sigma}^+\right) \quad \ldots \textit{decay of atom with rate } \Gamma$$

Free atomic Hamiltonian (e.g. with drive)

Lecture 2 - Lindblad master equations - Review

- General form of Lindblad master equation

$$\frac{d}{dt}\hat{\rho}_S = \mathcal{L}(\hat{\rho}_S) = -\mathrm{i}\left[\hat{H}_S, \hat{\rho}\right] + \sum_\eta \left(2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger - \hat{L}_\eta^\dagger \hat{L}_\eta \hat{\rho}_S - \hat{\rho}_S \hat{L}_\eta^\dagger \hat{L}_\eta\right)$$

- Alternative equivalent forms

$$\hat{H}_{\text{eff}} = \hat{H}_S - \mathrm{i}\sum_\eta \hat{L}_\eta^\dagger \hat{L}_\eta$$

*effective **non-hermitian** Hamiltonian*

non-trace preserving

$$\frac{d}{dt}\hat{\rho}_S = -\mathrm{i}\left(\hat{H}_{\text{eff}}\hat{\rho}_S - \hat{\rho}_S \hat{H}_{\text{eff}}^\dagger\right) + \sum_\eta 2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger$$

recycling term (fixes trace)

- In fact, the Lindblad form is the simplest way to write an linear evolution equation which is trace preserving

$$\mathrm{tr}\left[\frac{d}{dt}\hat{\rho}_S\right] = \mathrm{tr}\left\{\mathcal{L}(\hat{\rho}_S)\right\} = \sum_\eta \mathrm{tr}\left(2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger\right) - \mathrm{tr}\left(\hat{L}_\eta^\dagger \hat{L}_\eta \hat{\rho}_S\right) - \mathrm{tr}\left(\hat{\rho}_S \hat{L}_\eta^\dagger \hat{L}_\eta\right) = 0$$

cyclic invariance of trace

Lecture 2 - Lindblad master equation dynamics

- The right hand side of the master equation is just a function of the density matrix ("super-operator")

$$\frac{d}{dt}\hat{\rho}_S = \mathcal{L}(\hat{\rho}_S) = -\mathrm{i}\left[\hat{H}_S, \hat{\rho}\right] + \sum_\eta \left(2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger - \hat{L}_\eta^\dagger \hat{L}_\eta \hat{\rho}_S - \hat{\rho}_S \hat{L}_\eta^\dagger \hat{L}_\eta\right)$$

- We can simply define a statevector of matrix elements, and implement the super-operator as function

$$\mathbf{y} = [\rho_{1,1}, \rho_{2,1}, \rho_{3,1}, \ldots, \rho_{1,2}, \rho_{1,2}, \ldots]^T$$

$$\frac{d}{dt}\hat{\rho}_S = \mathcal{L}(\hat{\rho}_S) \quad \Leftrightarrow \quad \dot{\mathbf{y}}(t) = f(\mathbf{y}(t))$$

…we could plug this into Runge-Kutta!

- However, the **master equation** is also **linear!**

We can write it like $\dot{\mathbf{y}}(t) = f(\mathbf{y}(t)) = \mathbf{A} \cdot \mathbf{y}(t)$ *(next slide)*

Solution: $y(t) = \mathrm{e}^{t\mathbf{A}} y(0)$

If we construct the (sparse) matrix representing the whole "Liouvillian" matrix $\boldsymbol{A}$*, we can also use our Krylov space method again!*

Lecture 2 - Lindblad master equation dynamics

- How to construct the super-operator matrix? *Most common scheme*

$$\overset{D\times D}{\hat{\rho}=\begin{pmatrix}\rho_{1,1} & \rho_{1,2} & \rho_{1,3}\\ \rho_{2,1} & \rho_{2,2} & \rho_{2,3}\\ \rho_{3,1} & \rho_{3,2} & \rho_{3,3}\end{pmatrix}} \rightarrow \mathbf{y}=\big(\underbrace{\rho_{1,1},\rho_{2,1},\rho_{3,1}}_{\mathbf{y}_1^T},\underbrace{\rho_{1,2},\rho_{2,2},\rho_{3,2}}_{\mathbf{y}_2^T},\underbrace{\rho_{1,3},\rho_{2,3},\rho_{3,3}}_{\mathbf{y}_3^T}\big)^T$$

$$\underbrace{\begin{pmatrix}\mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3\end{pmatrix}}$$

$$\rho_{i,j} \quad \Leftrightarrow \quad y_v \qquad v=(j-1)D+i$$

- Now consider an operator $\hat{A}$

Kronecker product definition: $\hat{A}\otimes\hat{B}=\begin{pmatrix}A_{1,1}\hat{B} & A_{1,2}\hat{B} & \dots\\ A_{2,1}\hat{B} & A_{2,2}\hat{B} & \dots\\ \vdots & \vdots & \end{pmatrix}$

$$(\mathbb{1}\otimes\hat{A})\mathbf{y}=\underbrace{\begin{pmatrix}\hat{A} & 0 & 0\\ 0 & \hat{A} & 0\\ 0 & 0 & \hat{A}\end{pmatrix}}_{D^2\times D^2}\begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\end{pmatrix}=\hat{A}\begin{pmatrix}\mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3\end{pmatrix}=\hat{A}\hat{\rho}$$

Multiplication from left side

- Equally, it's easy to show *(exercise)*

$$(\hat{B}^T\otimes\mathbb{1})\mathbf{y}=\hat{\rho}\hat{B}$$

Multiplication from right side

$$(\hat{B}^T\otimes\hat{A})\mathbf{y}=\hat{A}\hat{\rho}\hat{B}$$

Multiplication from left and right side

Liouvillian matrix can be fully constructed with Kronecker products!

Lecture 2 - Lindblad master equation dynamics

$$\hat{\rho} = \begin{pmatrix} \rho_{1,1} & \rho_{1,2} & \rho_{1,3} \\ \rho_{2,1} & \rho_{2,2} & \rho_{2,3} \\ \rho_{3,1} & \rho_{3,2} & \rho_{3,3} \end{pmatrix} \rightarrow \mathbf{y} = \left(\rho_{1,1}, \rho_{2,1}, \rho_{3,1}, \rho_{1,2}, \rho_{2,2}, \rho_{3,2}, \rho_{1,3}, \rho_{2,3}, \rho_{3,3}\right)^T$$

```
rho = reshape(y, 2^N, 2^N)
```

```
y = rho[:] # memory ordering fits
```

$$(\mathbb{1} \otimes \hat{A})\mathbf{y} = \hat{A}\hat{\rho} \qquad (\hat{B}^T \otimes \mathbb{1})\mathbf{y} = \hat{\rho}\hat{B} \qquad (\hat{B}^T \otimes \hat{A})\mathbf{y} = \hat{A}\hat{\rho}\hat{B}$$

- Then:

$$\hat{H}_{\text{eff}} = \hat{H}_S - \mathrm{i}\sum_\eta \hat{L}_\eta^\dagger \hat{L}_\eta \qquad \frac{d}{dt}\hat{\rho}_S = -\mathrm{i}\left(\hat{H}_{\text{eff}}\hat{\rho}_S - \hat{\rho}_S \hat{H}_{\text{eff}}^\dagger\right) + \sum_\eta 2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger$$

$$-\mathrm{i}\left[(\mathbb{1} \otimes \hat{H}_{\text{eff}}) - (\hat{H}_{\text{eff}}^* \otimes \mathbb{1})\right]\mathbf{y}$$

element-wise conjugate

$$2(\hat{L}_\eta^* \otimes \hat{L}_\eta)\mathbf{y}$$

```
Heff = H
for nn = 1:N
    Heff -= 1im .* (gam/2) .* (sms[nn]' * sms[nn])
end
```

```
idD = sparse(I, 2^N, 2^N)
```

Example:

$$\hat{L}_\eta = \sqrt{\frac{\Gamma}{2}}\hat{\sigma}_\eta^-$$

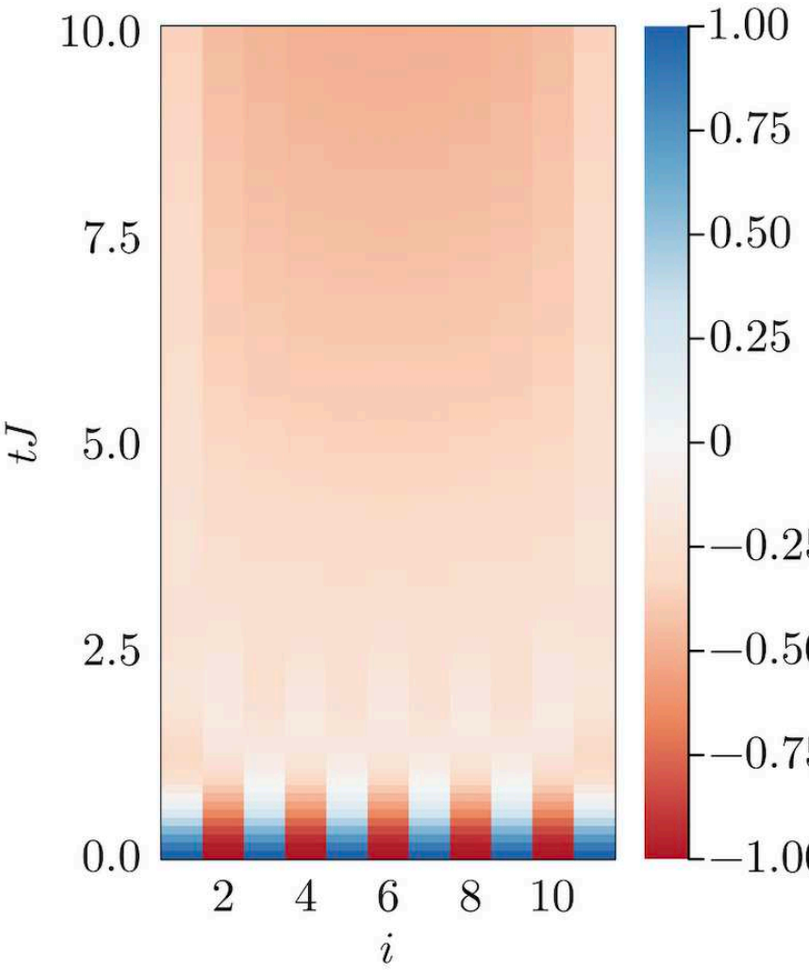
```
A = -1im .* (kron(idD, Heff) - kron(conj.(Heff), idD))

for nn = 1:N
    A += gam .* kron(sms[nn], sms[nn]) # sigmas are real
end
```

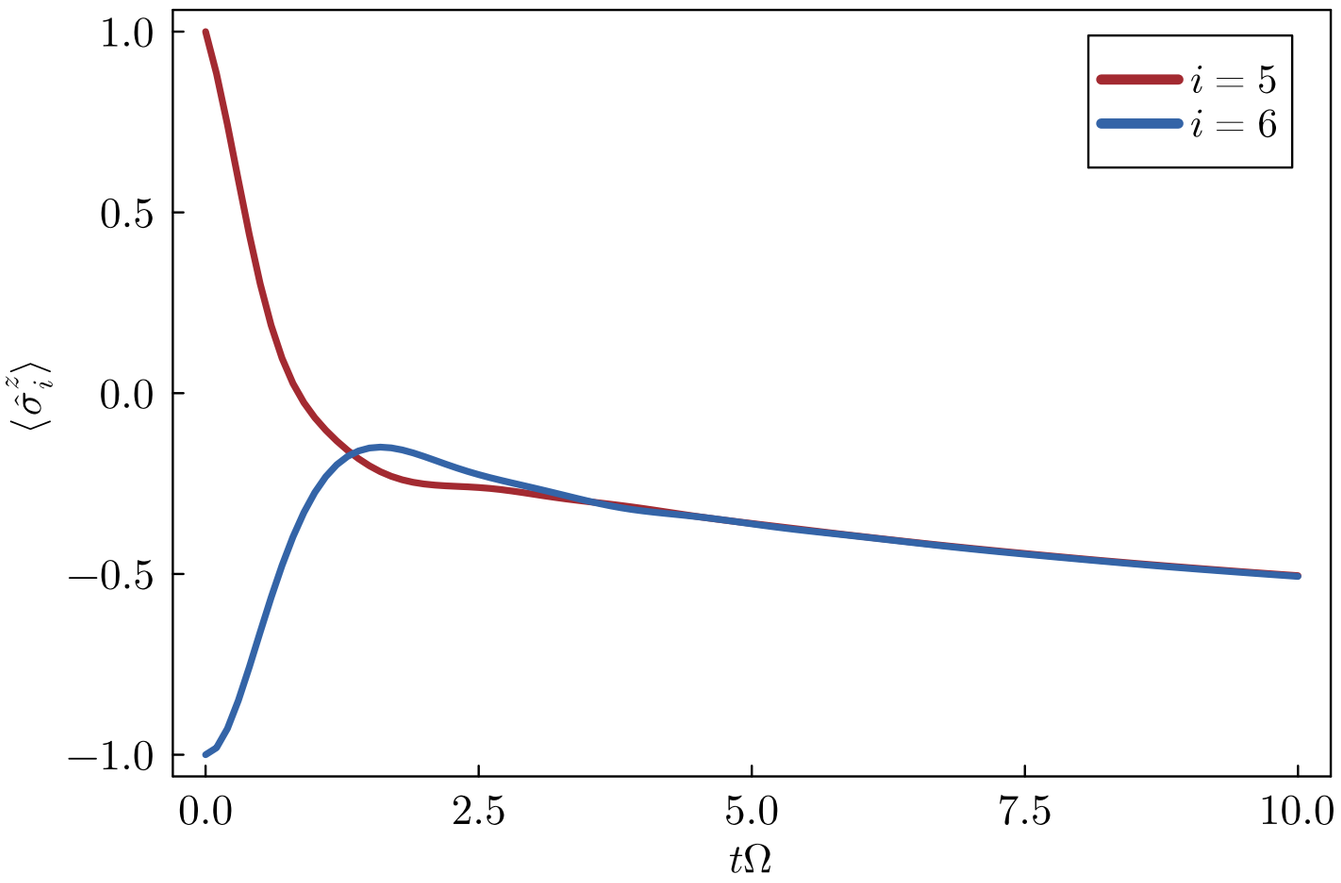
Lecture 2 - Lindblad master equation dynamics

- Evolution of local spin-z component
- *N = 11* spins

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad h_x = J$$

$$\alpha = 3 \qquad \Gamma = \frac{J}{2}$$





- **Remark:** The Liouvillian matrix construction is not ideal, an alternative, expand in Pauli basis

$$\hat{\rho} = \sum_{\{i_n\}=1}^{4} r_{i_1,i_2,\ldots,i_N}\,\hat{\sigma}_1^{r_1}\otimes\hat{\sigma}_1^{r_2}\otimes\ldots\,\hat{\sigma}_N^{r_N}$$

$$\hat{\sigma}_i^{0,1,2,3} = \{\mathbb{1}_i, \hat{\sigma}_i^x, \hat{\sigma}_i^y, \hat{\sigma}_i^z\}$$

$$\mathrm{tr}\left(\hat{\sigma}_i^n\hat{\sigma}_i^m\right) = \delta_{n,m}$$

Major advantage: The "Bloch" state vectors are real!

- **Remark:** Since we keep the exact full density matrix, we're now limited to ~half the size than before :(

Lecture 2 - Quantum Trajectories

- **Remark:** Since we keep the exact full density matrix, we're now limited to ~half the size than before :(
- We can fix that :)
- **Fundamentally:** Every density matrix is a statistical mixture of pure states: $\hat{\rho} = \sum_{\mu} p_{\mu} \left|\psi_{\mu}\right\rangle \left\langle\psi_{\mu}\right|$

"Quantum trajectories" $\Leftrightarrow$ *Let's evolve the state-vectors instead*

- The standard "Quantum Monte-Carlo wavefunction algorithm"

J. Dalibard, Y. Castin, and K. Mølmer, Phys. Rev. Lett. 68, 580 (1992); R. Dum, P. Zoller, and H. Ritsch, Phys. Rev. A 45, 4879 (1992)

$$\hat{H}_{\text{eff}} = \hat{H}_S - \mathrm{i}\sum_{\eta} \hat{L}_{\eta}^{\dagger}\hat{L}_{\eta} \qquad \frac{d}{dt}\hat{\rho}_S = -\mathrm{i}\left(\hat{H}_{\text{eff}}\hat{\rho}_S - \hat{\rho}_S\hat{H}_{\text{eff}}^{\dagger}\right) + \sum_{\eta} 2\hat{L}_{\eta}\hat{\rho}_S\hat{L}_{\eta}^{\dagger}$$

Small time-step Δt *Start in:* $\left|\psi_0\right\rangle$

1. Evolve the state-vector with the effective Hamiltonian. The norm will decay a bit, compute:

$$\left|\psi(\Delta t)\right\rangle = \mathrm{e}^{-\mathrm{i}\Delta t\hat{H}_{\text{eff}}}\left|\psi_0\right\rangle \qquad \Delta p = 1 - \left\langle\psi(\Delta t)|\psi(\Delta t)\right\rangle$$

2. Pick a uniformly distributed random number $r \in [0, 1]$

3a. If $r > \Delta p$, renormalize and go to **1** for next step $\left|\psi_0\right\rangle = \left|\psi\right\rangle(\Delta t)/\|\left|\psi(\Delta t)\right\rangle\|$

3b. Else, calculate probability distribution $\tilde{p}_{\eta} = \left\langle\psi_0\right|\hat{L}_{\eta}^{\dagger}\hat{L}_{\eta}\left|\psi_0\right\rangle \qquad p_{\eta} = p_{\eta}/\sum_{\eta}\tilde{p}_{\eta}$

Pick a jump operator μ accordingly and go to **1** with: $\left|\psi_0\right\rangle = \hat{L}_{\eta}\left|\psi_0\right\rangle / \|\hat{L}_{\eta}\left|\psi_0\right\rangle\|$

Lecture 2 - Quantum Trajectories

1. Evolve the state-vector with the effective Hamiltonian. The norm will decay a bit, compute:

$$|\psi(\Delta t)\rangle = \mathrm{e}^{-\mathrm{i}\Delta t \hat{H}_{\mathrm{eff}}} |\psi_0\rangle \qquad \Delta p = 1 - \langle\psi(\Delta t)|\psi(\Delta t)\rangle$$

2. Pick a uniformly distributed random number $r \in [0, 1]$

3a. If $r > \Delta p$, renormalize and go to **1** for next step $|\psi_0\rangle = |\psi\rangle\,(\Delta t)/\|\,|\psi(\Delta t)\rangle\,\|$

3b. Else, calculate probability distribution $\tilde{p}_\eta = \langle\psi_0|\,\hat{L}_\eta^\dagger \hat{L}_\eta\,|\psi_0\rangle \qquad p_\eta = p_\eta / \sum_\eta \tilde{p}_\eta$

Pick a jump operator μ accordingly and go to **1** with: $|\psi_0\rangle = \hat{L}_\eta\,|\psi_0\rangle\,/\|\hat{L}_\eta\,|\psi_0\rangle\,\|$

- Assume we have a pure state sample of $\hat{\rho}(t)$ at time t, $|\psi_0\rangle$ $\qquad \hat{H}_{\mathrm{eff}} = \hat{H}_S - \mathrm{i}\sum_\eta \hat{L}_\eta^\dagger \hat{L}_\eta$
- Then, after the time step, we have the averaged density matrix

$$\overline{\hat{\rho}(t+\Delta t)} = (1-\Delta p)\frac{|\psi(\Delta t)\rangle\,\langle\psi(\Delta t)|}{\|\,|\psi(\Delta t)\rangle\,\|^2} + \Delta p \sum_\eta p_\eta \frac{\hat{L}_\eta\,|\psi_0\rangle\,\langle\psi_0|\,\hat{L}_\eta^\dagger}{\|\hat{L}_\eta\,|\psi_0\rangle\,\|^2}$$

- In linear order: $\qquad |\psi(\Delta t)\rangle \approx |\psi_0\rangle - \mathrm{i}\hat{H}_{\mathrm{eff}}\Delta t\,|\psi_0\rangle \qquad \|\,|\psi(\Delta t)\rangle\,\|^2 = 1 - \Delta p$

$$\langle\psi(\Delta t)|\psi(\Delta t)\rangle \approx 1 - \mathrm{i}\Delta t\,\langle\psi_0|\,\hat{H}_{\mathrm{eff}}\,|\psi_0\rangle + \mathrm{i}\Delta t\,\langle\psi_0|\,\hat{H}_{\mathrm{eff}}^\dagger\,|\psi_0\rangle = 1 - 2\Delta t\sum_\eta \langle\psi_0|\,\hat{L}_\eta^\dagger \hat{L}_\eta\,|\psi_0\rangle$$

$$\Delta p = 2\Delta t \sum_\eta \tilde{p}_\eta \qquad p_\eta = \tilde{p}_\eta \frac{2\Delta t}{\Delta p}$$

Lecture 2 - Quantum Trajectories

1. Evolve the state-vector with the effective Hamiltonian. The norm will decay a bit, compute:

$$|\psi(\Delta t)\rangle = \mathrm{e}^{-\mathrm{i}\Delta t \hat{H}_{\mathrm{eff}}} |\psi_0\rangle \qquad \Delta p = 1 - \langle\psi(\Delta t)|\psi(\Delta t)\rangle$$

2. Pick a uniformly distributed random number $r \in [0, 1]$

3a. If $r > \Delta p$, renormalize and go to **1** for next step $|\psi_0\rangle = |\psi\rangle\,(\Delta t)/\|\,|\psi(\Delta t)\rangle\,\|$

3b. Else, calculate probability distribution $\tilde{p}_\eta = \langle\psi_0|\,\hat{L}_\eta^\dagger \hat{L}_\eta\,|\psi_0\rangle \qquad p_\eta = p_\eta / \sum_\eta \tilde{p}_\eta$

Pick a jump operator μ accordingly and go to **1** with: $|\psi_0\rangle = \hat{L}_\eta\,|\psi_0\rangle\,/\|\hat{L}_\eta\,|\psi_0\rangle\,\|$

- Assume we have a pure state sample of $\hat{\rho}(t)$ at time t, $|\psi_0\rangle$ $\qquad \hat{H}_{\mathrm{eff}} = \hat{H}_S - \mathrm{i}\sum_\eta \hat{L}_\eta^\dagger \hat{L}_\eta$
- Then, after the time step, we have the averaged density matrix

$$\overline{\hat{\rho}(t+\Delta t)} = (1-\Delta p)\frac{|\psi(\Delta t)\rangle\,\langle\psi(\Delta t)|}{\|\,|\psi(\Delta t)\rangle\,\|^2} + \Delta p \sum_\eta p_\eta \frac{\hat{L}_\eta\,|\psi_0\rangle\,\langle\psi_0|\,\hat{L}_\eta^\dagger}{\|\hat{L}_\eta\,|\psi_0\rangle\,\|^2} \qquad p_\eta = \tilde{p}_\eta \frac{2\Delta t}{\Delta p}$$

- In linear order:

$$\overline{\hat{\rho}(t+\Delta t)} = |\psi(\Delta t)\rangle\,\langle\psi(\Delta t)| + 2\Delta t \sum_\eta \hat{L}_\eta\,|\psi_0\rangle\,\langle\psi_0|\,\hat{L}_\eta^\dagger$$

This is precisely the evolution of the Lindblad master equation!

- **Warning:** This is in linear order, time-steps need to be small, meaning $\Delta t \Gamma \ll 1$

… often gamma is smaller than coherent rates, then higher order integration for time-step still makes sense

Lecture 2 - Quantum Trajectories

1. Evolve the state-vector with the effective Hamiltonian. The norm will decay a bit, compute:

$$|\psi(\Delta t)\rangle = \mathrm{e}^{-\mathrm{i}\Delta t \hat{H}_{\mathrm{eff}}} |\psi_0\rangle \qquad \Delta p = 1 - \langle\psi(\Delta t)|\psi(\Delta t)\rangle$$

2. Pick a uniformly distributed random number $r \in [0, 1]$

3a. If $r > \Delta p$, renormalize and go to **1** for next step $|\psi_0\rangle = |\psi\rangle\,(\Delta t)/\|\,|\psi(\Delta t)\rangle\,\|$

3b. Else, calculate probability distribution $\tilde{p}_\eta = \langle\psi_0|\,\hat{L}_\eta^\dagger \hat{L}_\eta\,|\psi_0\rangle \qquad p_\eta = p_\eta / \sum_\eta \tilde{p}_\eta$

Pick a jump operator μ accordingly and go to **1** with: $|\psi_0\rangle = \hat{L}_\eta\,|\psi_0\rangle\,/\|\hat{L}_\eta\,|\psi_0\rangle\,\|$

```
using Random
rng = MersenneTwister(1907)
```

Pseudo-random number seed

- Run the full evolution for N_t trajectories

$$\hat{\rho} \approx \frac{1}{N_t} \sum_{\nu=1}^{N_t} |\psi_\nu\rangle\,\langle\psi_\nu|$$

- **Note:** Expectation values - simple mean value over trajectories

$$\mathrm{tr}\left(\hat{O}\hat{\rho}\right) \approx \frac{1}{N_t} \sum_{\nu=1}^{N_t} \langle\psi_\nu|\,\hat{O}\,|\psi_\nu\rangle$$

```
psit, info = exponentiate(Heff, -1im * dt, psi)

nrm = norm(psit)
if nrm > rand(rng)  # no jump
    psi = psit ./ nrm
else    # do jump
    for nn = 1:N
        peta[nn] = real(psi' * LLvec[nn] * psi)
    end
    peta = peta ./ sum(peta)
    mu = findfirst(cumsum(peta) .> rand(rng))
    psi = sms[mu]*psi
    psi ./= norm(psi)
end
```

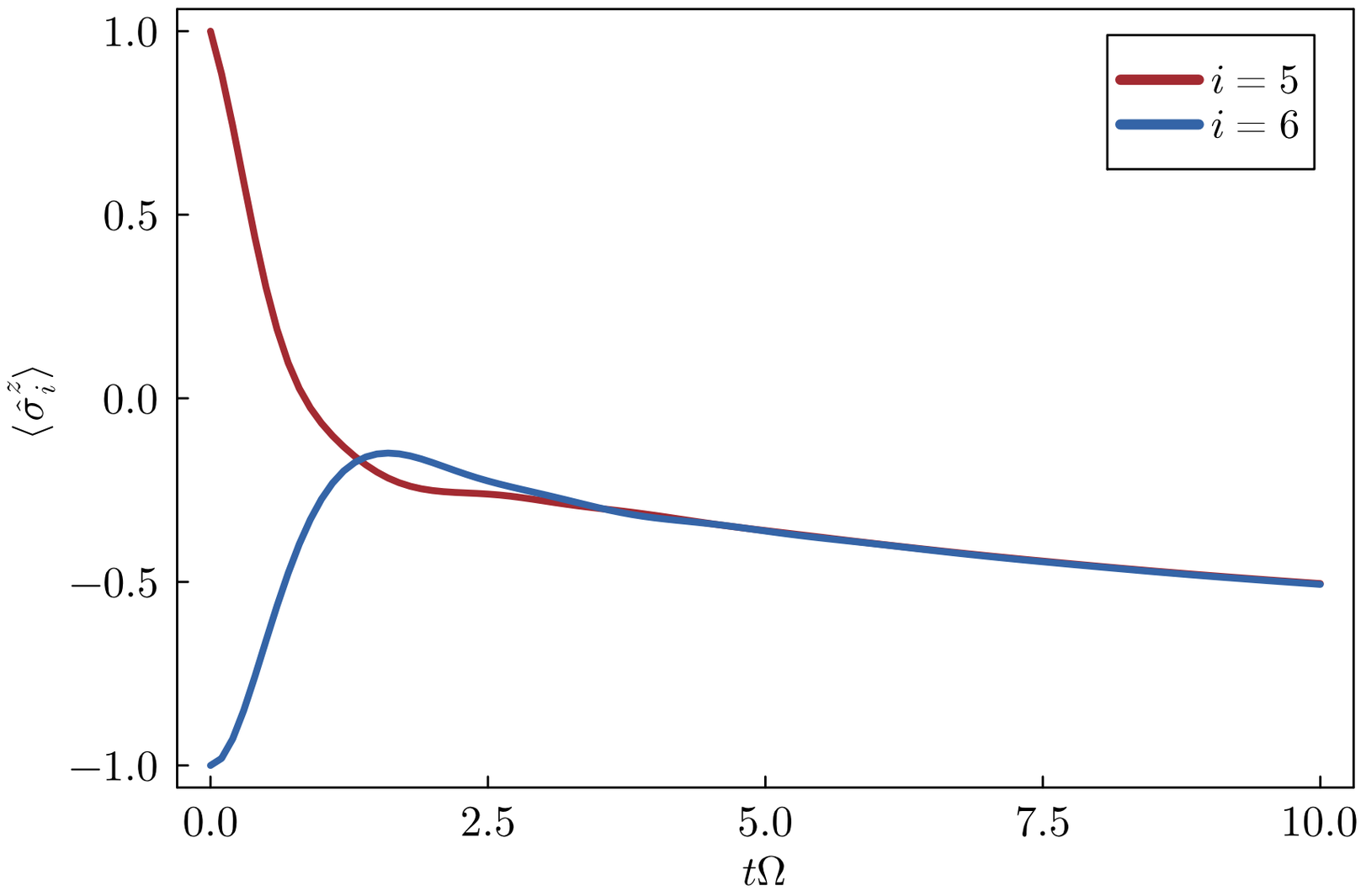
Lecture 2 - Lindblad master equation dynamics

- Evolution of local spin-z component
- *N = 11* spins

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad h_x = J$$

$$\alpha = 3 \qquad \Gamma = \frac{J}{2}$$

- Reminder - **full density matrix:**



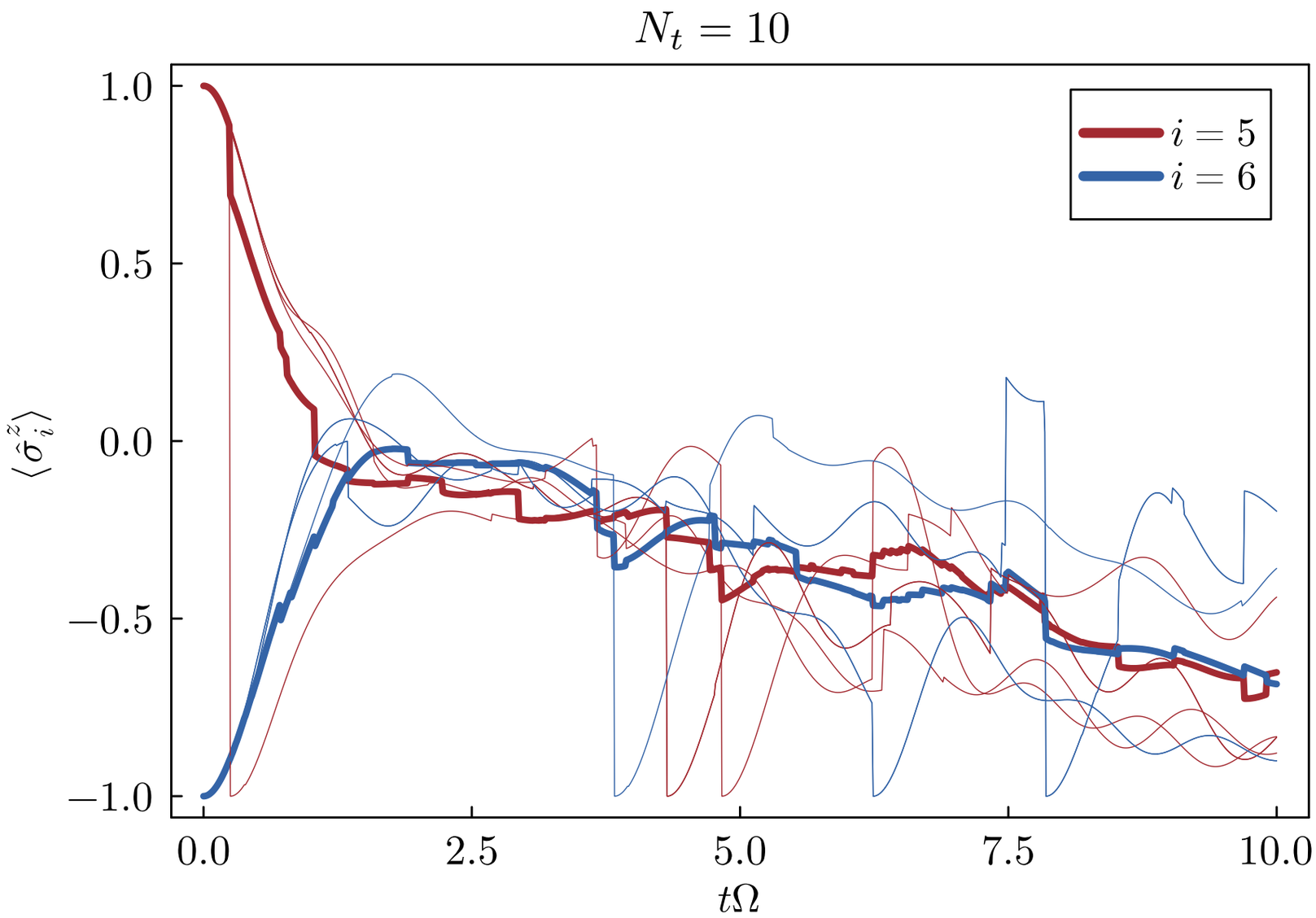
Lecture 2 - Lindblad master equation dynamics

- Evolution of local spin-z component
- *N = 11* spins

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad h_x = J$$

$$\alpha = 3 \qquad \Gamma = \frac{J}{2}$$

- **Trajectories:**



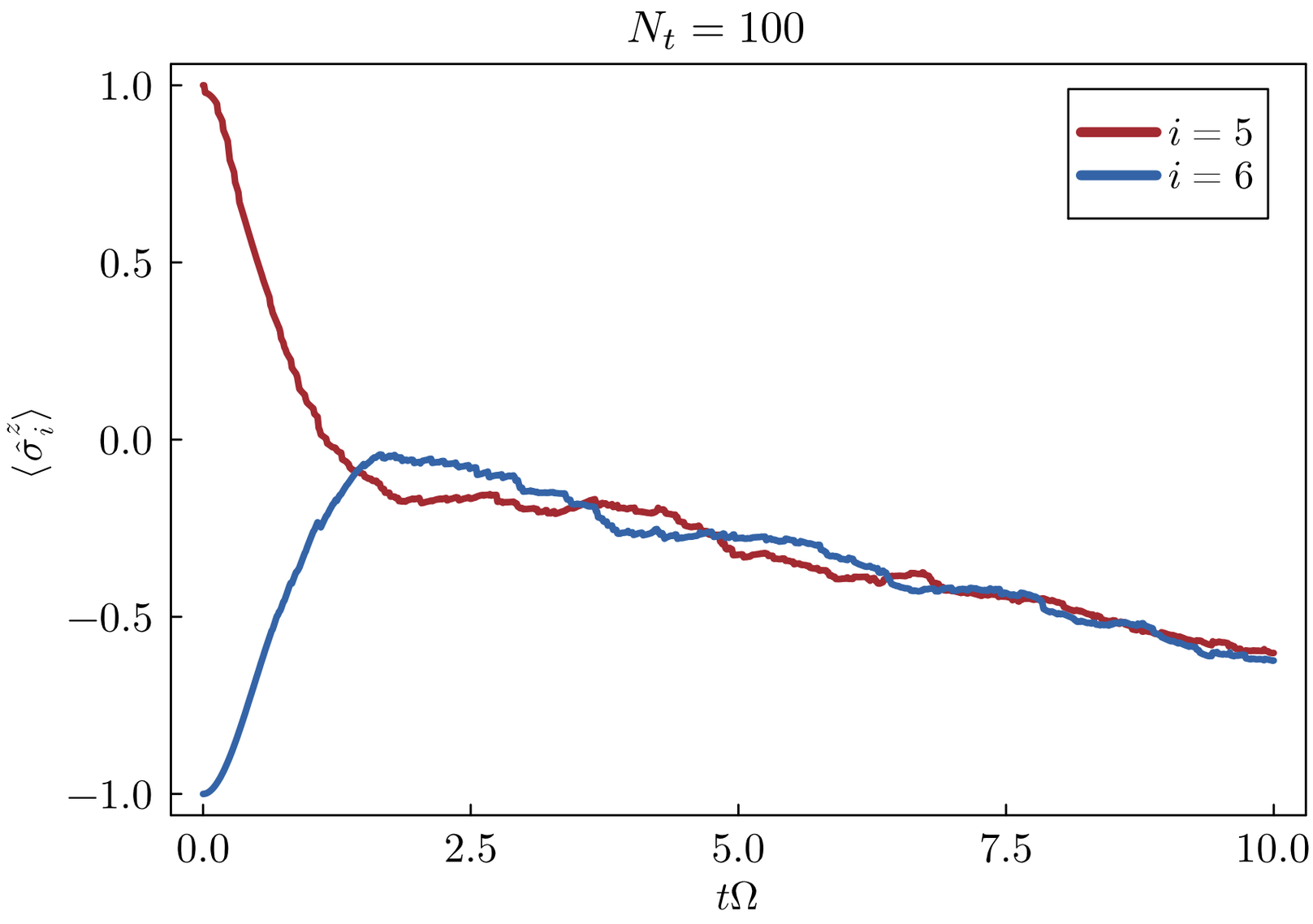
Lecture 2 - Lindblad master equation dynamics

- Evolution of local spin-z component
- *N = 11* spins

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad h_x = J$$

$$\alpha = 3 \qquad \Gamma = \frac{J}{2}$$

- **Trajectories:**



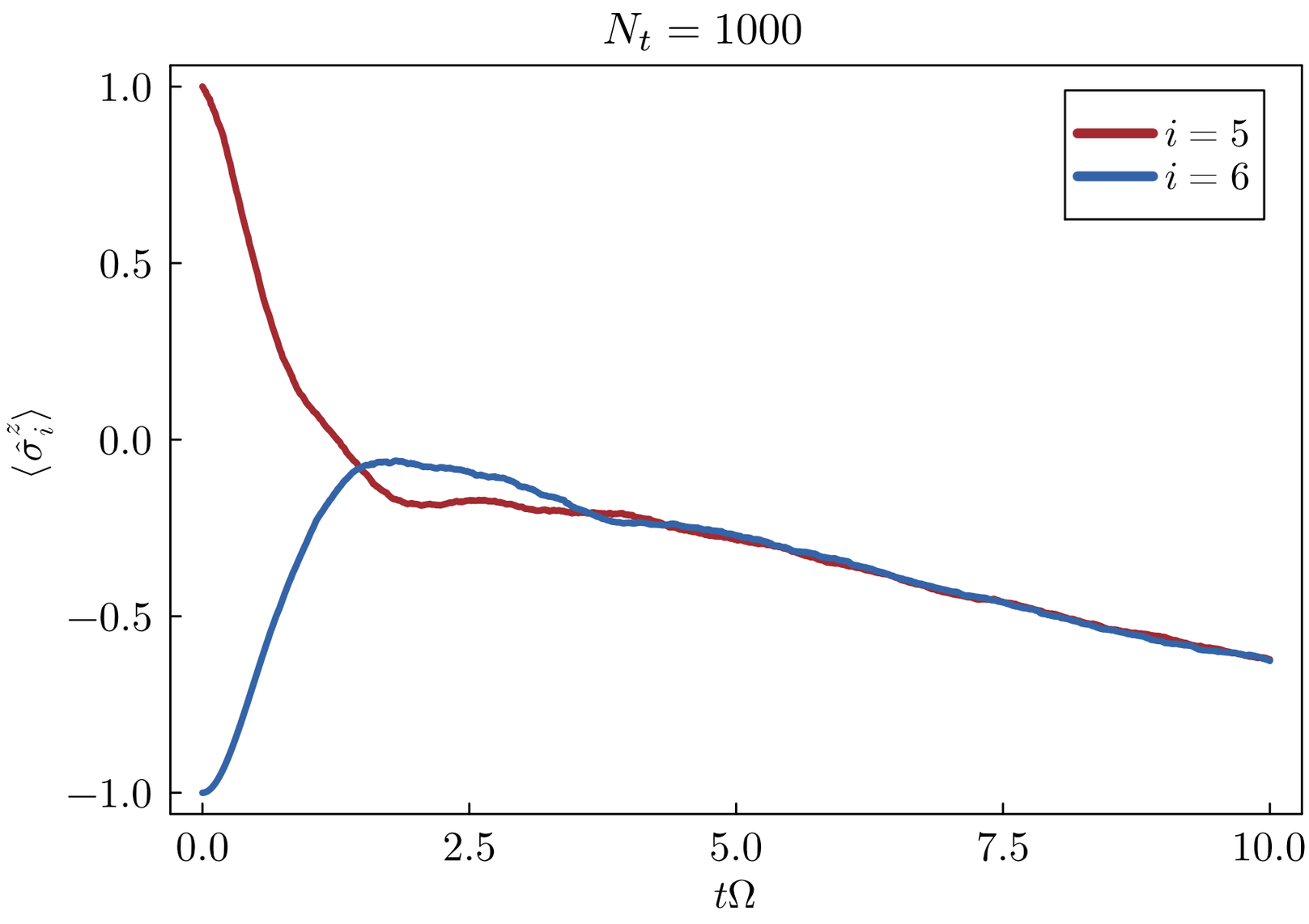
Lecture 2 - Lindblad master equation dynamics

- Evolution of local spin-z component
- *N = 11* spins

$$\hat{H}_{\mathrm{TI}} = \sum_{i<j} J_{i,j}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x \qquad J_{ij} = \frac{J}{|i-j|^\alpha} \qquad h_x = J$$

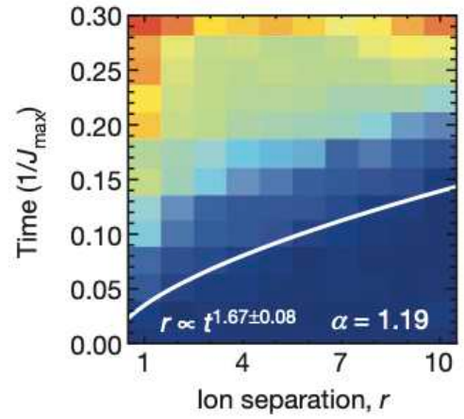
$$\alpha = 3 \qquad \Gamma = \frac{J}{2}$$

- **Trajectories:**



- **Remark:** This only works for simple observables. For correlations that are small, **many trajectories** needed. Statistical error bars only decrease as $1/\sqrt{N_t}$
- **On the other hand:** Many trajectories are easy to compute on clusters, since the method parallelizes trivially!

Lecture 2 - Recap

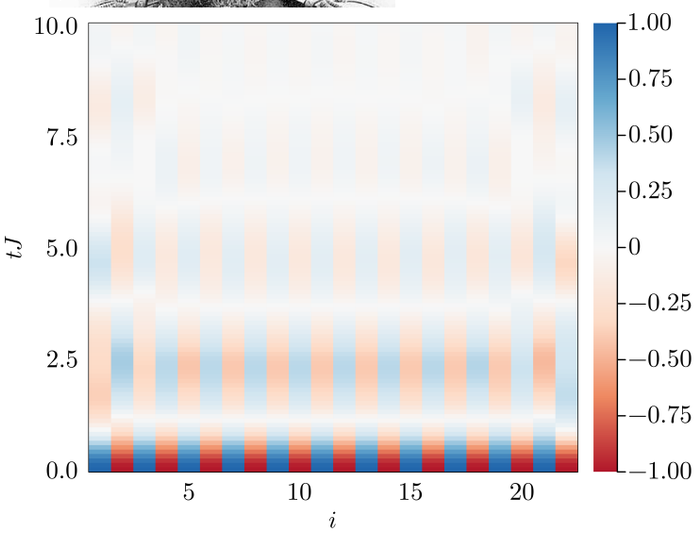


- We discussed some general spin-model physics with long-range couplings, as they can e.g. be engineered in trapped ion systems. We discussed how to construct Hamiltonians using Kronecker products. It's very important to use **sparse matrices**.

$$\hat{\sigma}_i^- = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \end{pmatrix}$$

- We introduced a new time-evolution algorithm for linear systems, based on: **Krylov space**. Krylov space is a vector-space constructed from an initial state and the evolution matrix:

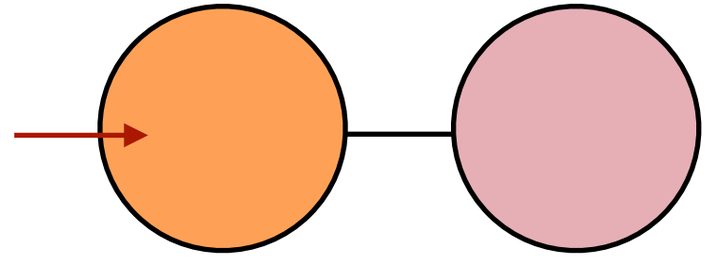
$$\text{span}\left(\hat{A}^0 |\psi_0\rangle, \hat{A}^1 |\psi_0\rangle, \hat{A}^2 |\psi_0\rangle, \ldots, \hat{A}^{m-1} |\psi_0\rangle\right)$$



Eigenvectors need to be made orthonormal, using **Arnoldi** or **Lanczos** iterations, then diagonalizations and matrix exponentials can be performed very efficiently on the much smaller **Krylov space.** This allows to easily simulate quantum dynamics of ~22 spins/qubits on a laptop.

- Finally we discussed how to simulate open system dynamics of **Lindblad master equations:**

$$\frac{d}{dt}\hat{\rho}_S = \mathcal{L}(\hat{\rho}_S) = -\mathrm{i}\left[\hat{H}_S, \hat{\rho}\right] + \sum_\eta \left(2\hat{L}_\eta \hat{\rho}_S \hat{L}_\eta^\dagger - \hat{L}_\eta^\dagger \hat{L}_\eta \hat{\rho}_S - \hat{\rho}_S \hat{L}_\eta^\dagger \hat{L}_\eta\right)$$



… either by **vectorization of the density matrix**, or by **quantum trajectories** (an unravelling of the density matrix into stochastic pure state evolutions)