Hands-on advanced numerical methods for quantum many-body dynamics

A tour through numerical methods for simulating large-scale quantum physics (classically)

Motto: Do it from scratch to better understand quantum physics and classical limitations

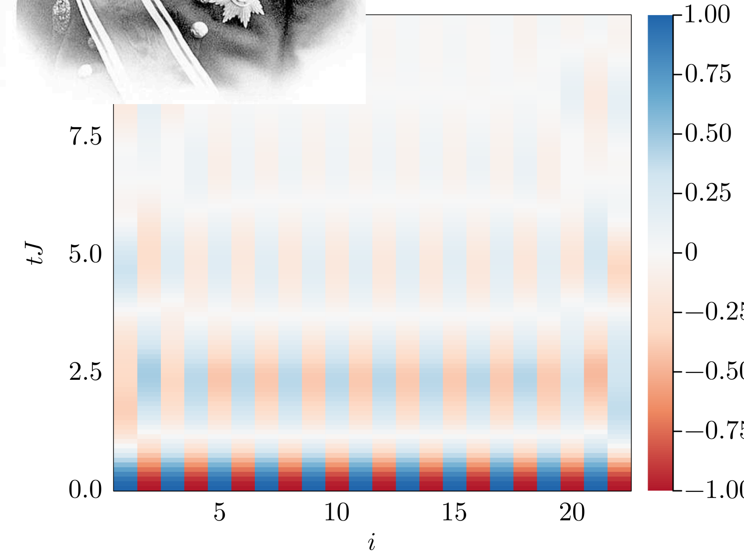
Structure: Theory lecture part + tutorial-style

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

- **Outline** (may vary)

1. **20/10:** Basic concepts: Numbers on computers, basic exact diagonalization (ED)
   Tutorial: ED on a simple spin model
2. **21/10:** A better ED, sparse matrices, Krylov space. Open systems.
   Tutorial: Spin-model simulations using Krylov space
3. **22/10**: Mean-field, Runge-Kutta
   Tutorial: A mean-field simulation of the transverse Ising model
4. **23/10:** How to go beyond: Matrix product states

Recap



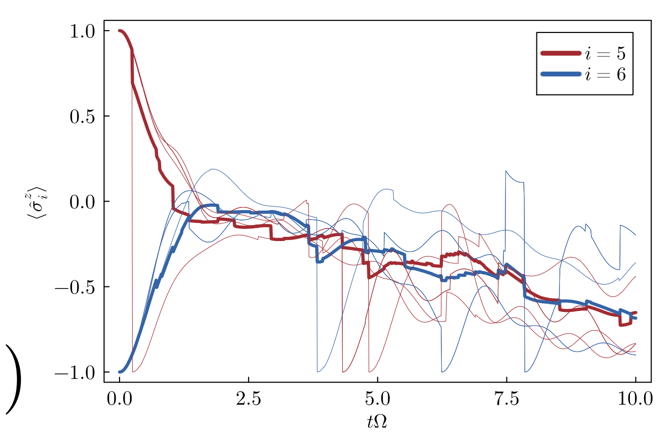
- We introduced a time-evolution algorithm for linear systems, based on: **Krylov space**. Krylov space is a vector-space constructed from an initial state and the evolution matrix:

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

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.

- We used Krylov space (Arnoldi exponentiation) to simulate correlation dynamics in long-range spin-models.

- Open system dynamics: Use **full density matrix simulations** or **Quantum trajectories**



$$D \times D$$

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

$$(\mathbf{y}_1 \quad \mathbf{y}_2 \quad \mathbf{y}_3)$$

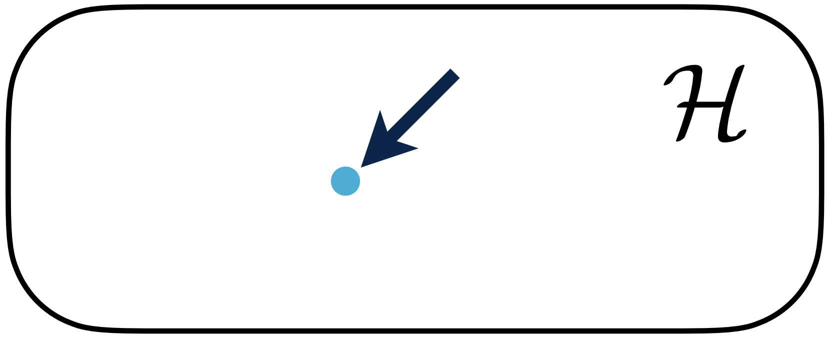
Master equation is linear, use Krylov space

- Quantum trajectories can be easily derived and simulated using the Kraus operator formalism

This time

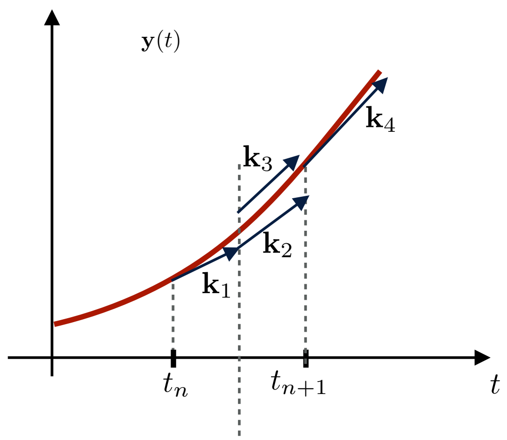
- **Part 1:** Mean-field approach to spin-models

$$|\psi\rangle \approx |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_N\rangle$$

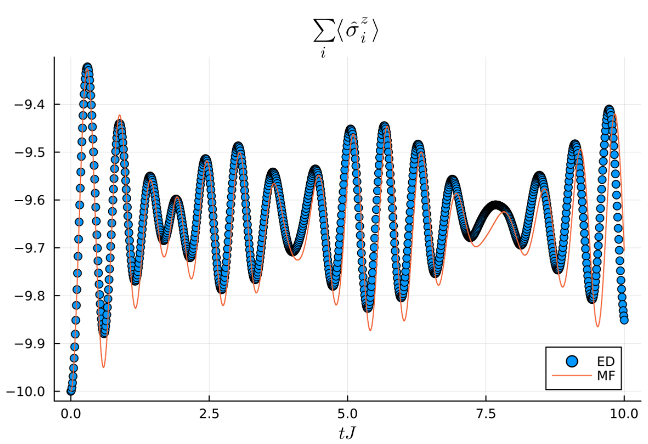


- **Part 2:** Runge-Kutta (RK) time-evolution methods: A swiss army knife

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



- **Tutorial:** A mean-field simulation of the transverse Ising model



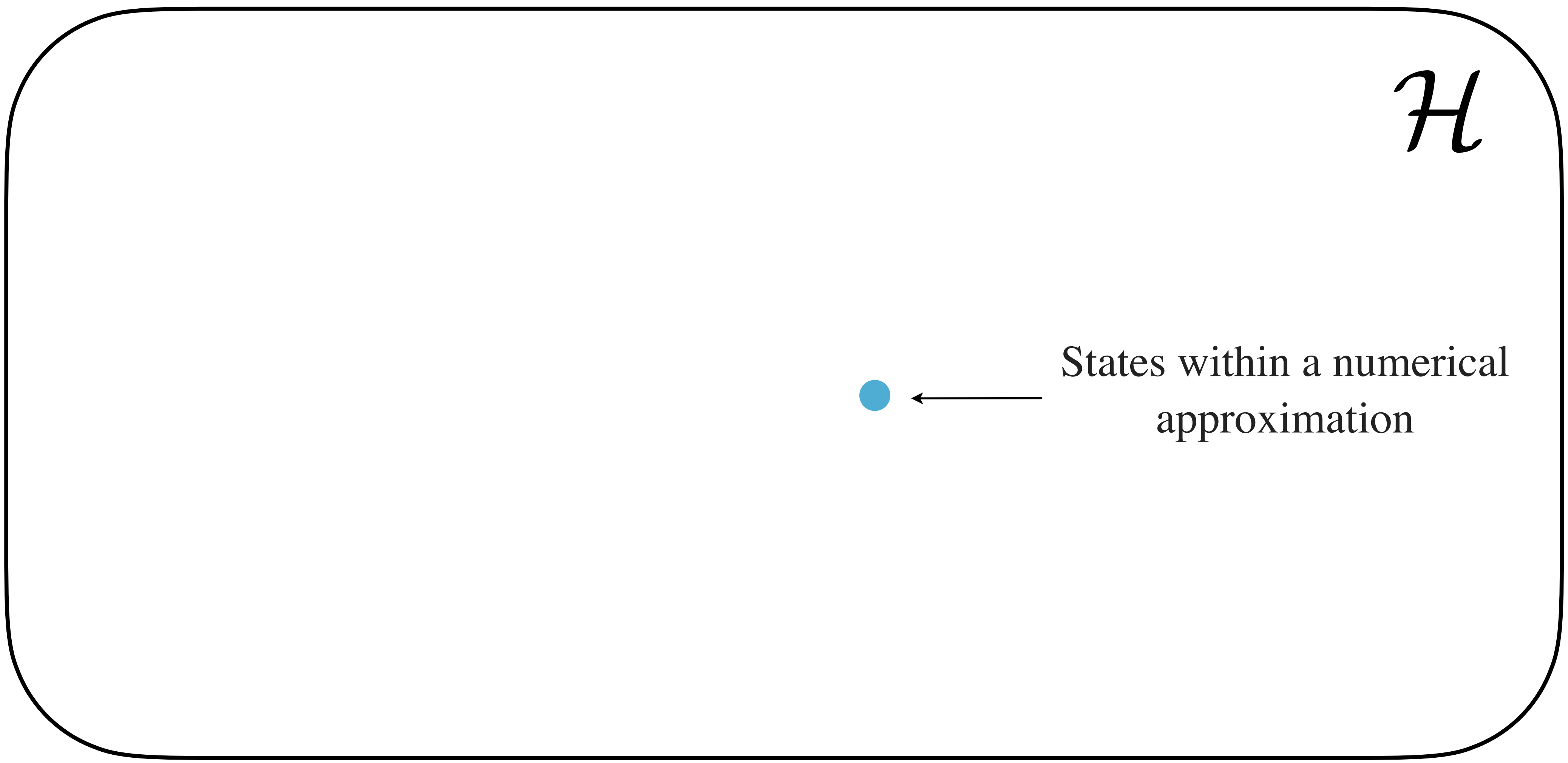
- **Part 3:** Mean-field dynamics for bosonic systems (the Gross-Pitaevskii equation)

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

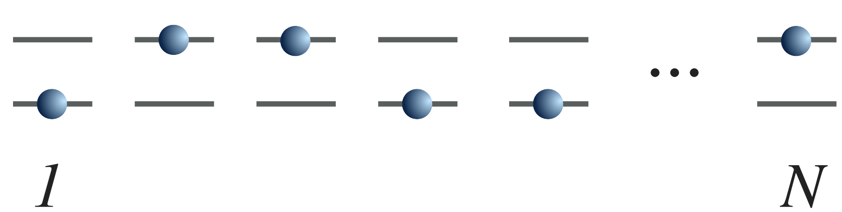
Mean-field approach to spin-models

- So far: we only considered **exact simulations on the full Hilbert space!**
- Now: Let’s **approximate the state** (reduce the effective Hilbert space size)

Gigantic Hilbert space

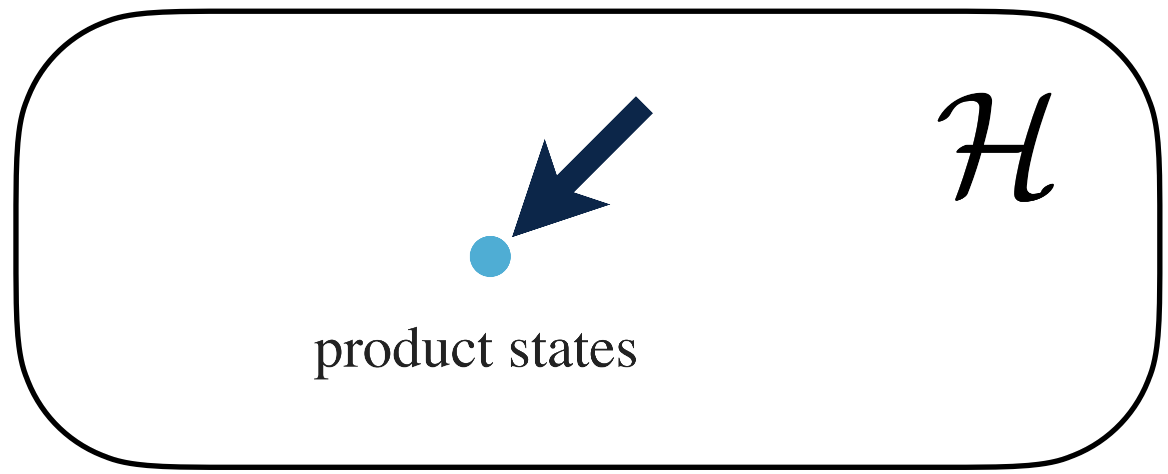


- For N spin-1/2 particles:
  $d = 2$

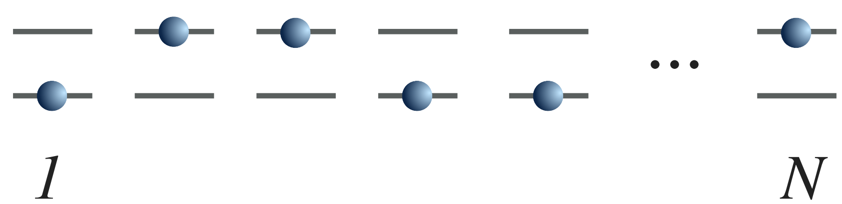


$$\dim(\mathcal{H}) = d^N = 2^N$$

Mean-field approach to spin-models



- For N spin-1/2 particles:
  $d = 2$



$$\dim(\mathcal{H}) = d^N = 2^N$$

$$|\psi\rangle = \sum_{i_1, i_2, \ldots, i_N = 1}^{d} c_{i_1, i_2, \ldots, i_N} |i_1, i_2, \ldots i_N\rangle$$

- "**Mean-field corner**" of the Hilbert space: **Product states** only (**neglect entanglement**)

$$|\psi\rangle \approx |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_N\rangle \qquad c_{i_1, i_2, \ldots, i_N} \approx c^{[1]}_{i_1} c^{[2]}_{i_2}, \ldots, c^{[N]}_{i_N}$$

- **Radical approximation:** Number of complex amplitudes N vs. 2^N

Mean-field approach to spin-models

- Let's derive equations of motion:

- **Strategy:** Derive equation of motion for the spin components of each spin i

$$\hat{\rho}_i = \frac{1}{2}\left(\mathbb{1} + \mathbf{r}_i \cdot \hat{\boldsymbol{\sigma}}\right) = \left(\mathbb{1} + x_i\hat{\sigma}_i^x + y_i\hat{\sigma}_i^y + z_i\hat{\sigma}_i^z\right)$$

$$x_i(t) \equiv \langle \hat{\sigma}_i^x(t) \rangle$$
$$y_i(t) \equiv \langle \hat{\sigma}_i^y(t) \rangle$$
$$z_i(t) \equiv \langle \hat{\sigma}_i^z(t) \rangle$$

- Use Heisenberg equations of motion to obtain: $\frac{d}{dt}\langle \hat{\sigma}_i^x \rangle = \mathrm{i}\left\langle \left[\hat{H}, \hat{\sigma}_i^x\right] \right\rangle$

… then enforce factorizations e.g. of the form: $\langle \hat{\sigma}_i^x \hat{\sigma}_i^y \rangle \approx \langle \hat{\sigma}_i^x \rangle \langle \hat{\sigma}_i^y \rangle$ *on the RHS*

- Let's derive it for a generic two-body Hamiltonian of the form:

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z \qquad V_{ii}^{\alpha\alpha} = 0 \quad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

Mean-field approach to spin-models

- A generic two-body spin-model Hamiltonian

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z \qquad V_{ii}^{\alpha\alpha} = 0 \quad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

- **Single-body terms:**

$$\frac{d}{dt}\hat{\sigma}_m^x = \mathrm{i}\left[\hat{H}_1, \hat{\sigma}_m^x\right] = \mathrm{i}b_m^y\left[\hat{\sigma}_m^y, \hat{\sigma}_m^x\right] + \mathrm{i}b_m^z\left[\hat{\sigma}_m^z, \hat{\sigma}_m^x\right] = 2b_m^y\hat{\sigma}_m^z - 2b_m^z\hat{\sigma}_m^y$$

using: $(x, y, z) = (1, 2, 3)$

$$\left[\hat{\sigma}^\alpha, \hat{\sigma}^\beta\right] = 2\mathrm{i}\sum_\gamma \epsilon_{\alpha\beta\gamma}\hat{\sigma}^\gamma$$

Levi-Civita tensor

$$\epsilon_{123} = \epsilon_{312} = \epsilon_{231} = 1$$

$$\epsilon_{321} = \epsilon_{132} = \epsilon_{213} = -1$$

0 otherwise

- Factorization is trivial $\quad x_i(t) \equiv \langle\hat{\sigma}_i^x(t)\rangle \quad y_i(t) \equiv \langle\hat{\sigma}_i^y(t)\rangle \quad z_i(t) \equiv \langle\hat{\sigma}_i^z(t)\rangle$

$$\frac{d}{dt}x_m = 2b_m^y z_m - 2b_m^z y_m$$

Mean-field approach to spin-models

- A generic two-body spin-model Hamiltonian

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z \qquad V_{ii}^{\alpha\alpha} = 0 \quad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

- **Single-body terms:**

- **Remark:** These are identical to fully classical spin equations (alternative derivation)

Poisson bracket for spin-dynamics

$$\dot{\alpha}_m = \{\alpha_m, H_C\} = 2\sum_\beta \epsilon_{\alpha\beta\gamma}\gamma_m \frac{\partial H_C}{\partial \beta_m} \qquad H_C(x_m, y_m, z_m) = b_m^x x_m + b_m^y y_m + b_m^z z_m$$

$$\dot{x}_m = \{x_m, H_C\} = 2\epsilon_{xyz}z_m\frac{\partial H_C}{\partial y_m} + 2\epsilon_{xzy}y_m\frac{\partial H_C}{\partial z_m} = 2z_m b_m^y - 2y_m b_m^z \quad ✓$$

- **Other equations:**

$$\frac{d}{dt}y_m = -2b_m^x z_m + 2b_m^z x_m$$

$$\frac{d}{dt}z_m = 2b_m^x y_m - 2b_m^y x_m$$

(exercise)

$(x, y, z) = (1, 2, 3)$

Levi-Civita tensor

$\epsilon_{123} = \epsilon_{312} = \epsilon_{231} = 1$

$\epsilon_{321} = \epsilon_{132} = \epsilon_{213} = -1$

0 otherwise

Mean-field approach to spin-models

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z$$

$$V_{ii}^{\alpha\alpha} = 0 \qquad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

- **Two-body terms:**

$$\frac{d}{dt}\hat{\sigma}_m^x = \mathrm{i}\left[\hat{H}_2, \hat{\sigma}_m^x\right]$$

$$= \frac{\mathrm{i}}{2}\left(\sum_j V_{mj}^{yy}[\hat{\sigma}_m^y, \hat{\sigma}_m^x]\hat{\sigma}_j^y + \sum_i V_{im}^{yy}\hat{\sigma}_i^y[\hat{\sigma}_m^y, \hat{\sigma}_m^x] + \sum_j V_{ij}^{zz}[\hat{\sigma}_m^z, \hat{\sigma}_m^x]\hat{\sigma}_j^z + \sum_i V_{mj}^{zz}\hat{\sigma}_i^z[\hat{\sigma}_m^z, \hat{\sigma}_m^x]\right)$$

$V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$

$$= -\sum_j V_{mj}^{yy}(-\hat{\sigma}_m^z\hat{\sigma}_j^y - \hat{\sigma}_j^y\hat{\sigma}_m^z) - \sum_j V_{mj}^{zz}(\hat{\sigma}_m^y\hat{\sigma}_j^z + \hat{\sigma}_j^z\hat{\sigma}_m^y)$$

$V_{ii}^{\alpha\alpha} = 0$

$$= 2\sum_j V_{mj}^{yy}\hat{\sigma}_m^z\hat{\sigma}_j^y - 2\sum_j V_{mj}^{zz}\hat{\sigma}_m^y\hat{\sigma}_j^z$$

using: $(x, y, z) = (1, 2, 3)$

$$\left[\hat{\sigma}^\alpha, \hat{\sigma}^\beta\right] = 2\mathrm{i}\sum_\gamma \epsilon_{\alpha\beta\gamma}\hat{\sigma}^\gamma$$

Levi-Civita tensor

$$\epsilon_{123} = \epsilon_{312} = \epsilon_{231} = 1$$
$$\epsilon_{321} = \epsilon_{132} = \epsilon_{213} = -1$$

0 otherwise

- Factorization: $x_i(t) \equiv \langle\hat{\sigma}_i^x(t)\rangle \quad y_i(t) \equiv \langle\hat{\sigma}_i^y(t)\rangle \quad z_i(t) \equiv \langle\hat{\sigma}_i^z(t)\rangle$

$$\frac{d}{dt}x_m \approx 2\sum_j V_{mj}^{yy}\langle\hat{\sigma}_m^z\rangle\langle\hat{\sigma}_j^y\rangle - 2\sum_j V_{mj}^{zz}\langle\hat{\sigma}_m^y\rangle\langle\hat{\sigma}_j^z\rangle$$

$$= z_m\left(2\sum_j V_{mj}^{yy}y_j\right) - y_m\left(2\sum_j V_{mj}^{zz}z_j\right)$$

Mean-field approach to spin-models

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z$$

$$V_{ii}^{\alpha\alpha} = 0 \qquad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

- **Two-body terms:**

$$\frac{d}{dt}x_m \approx z_m\left(2\sum_j V_{mj}^{yy}y_j\right) - y_m\left(2\sum_j V_{mj}^{zz}z_j\right)$$

$$\frac{d}{dt}y_m \approx -z_m\left(2\sum_j V_{mj}^{xx}x_j\right) + x_m\left(2\sum_j V_{mj}^{zz}z_j\right) \qquad \textit{(exercise)}$$

$$\frac{d}{dt}z_m \approx -x_m\left(2\sum_j V_{mj}^{yy}y_j\right) + y_m\left(2\sum_j V_{mj}^{xx}x_j\right) \qquad \textit{(exercise)}$$

- **Why is it called "mean-field"?**

$$\frac{d}{dt}x_m \approx z_m\Big(\underbrace{2\sum_j V_{mj}^{yy}y_j}_{\text{“mean-field”} \parallel y}\Big) - y_m\Big(\underbrace{2\sum_j V_{mj}^{zz}z_j}_{\text{“mean-field”} \parallel z}\Big)$$

- The mean-fields created by the other spins lead to a classical precession of spin *m*

Mean-field approach to spin-models

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z$$

$$V_{ii}^{\alpha\alpha} = 0 \qquad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

- **Two-body terms:**

$$\frac{d}{dt}x_m \approx z_m\left(2\sum_j V_{mj}^{yy}y_j\right) - y_m\left(2\sum_j V_{mj}^{zz}z_j\right)$$

$$\frac{d}{dt}y_m \approx -z_m\left(2\sum_j V_{mj}^{xx}x_j\right) + x_m\left(2\sum_j V_{mj}^{zz}z_j\right)$$

$$\frac{d}{dt}z_m \approx -x_m\left(2\sum_j V_{mj}^{yy}y_j\right) + y_m\left(2\sum_j V_{mj}^{xx}x_j\right)$$

- **Remark:** Note that all equations can also again be derived from the full classical ansatz, e.g.

$$\dot{\alpha}_m = \{\alpha_m, H_C\} = 2\sum_\beta \epsilon_{\alpha\beta\gamma}\gamma_m\frac{\partial H_C}{\partial\beta_m} \qquad H_C(\{x_i, y_i, z_i\}) = \sum_{ij}V_{ij}^{zz}z_iz_j$$

$$\dot{x}_m = 2\epsilon_{xzy}y_m\frac{\partial H_C}{\partial z_m} + 2\epsilon_{xyz}z_m\frac{\partial H_C}{\partial y_m}$$

$$= -2y_m\frac{1}{2}\left(\sum_i V_{im}^{zz}z_j + \sum_j V_{mj}^{zz}z_i\right) = -y_m\left(2\sum_j V_{mj}^{zz}\right) \checkmark$$

$$(x, y, z) = (1, 2, 3)$$

Levi-Civita tensor

$$\epsilon_{123} = \epsilon_{312} = \epsilon_{231} = 1$$

$$\epsilon_{321} = \epsilon_{132} = \epsilon_{213} = -1$$

0 otherwise

Mean-field approach to spin-models

$$\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i \mathbf{b}_i \cdot \boldsymbol{\sigma}_i + \frac{1}{2}\sum_{ij}\left(V_{ij}^{xx}\hat{\sigma}_i^x\hat{\sigma}_j^x + V_{ij}^{yy}\hat{\sigma}_i^y\hat{\sigma}_j^y + V_{ij}^{zz}\hat{\sigma}_i^z\hat{\sigma}_j^z\right)$$

$$\mathbf{b}_i \cdot \boldsymbol{\sigma}_i = b_i^x\hat{\sigma}_i^x + b_i^y\hat{\sigma}_i^y + b_i^z\hat{\sigma}_i^z$$

$$V_{ii}^{\alpha\alpha} = 0 \qquad V_{ij}^{\alpha\alpha} = V_{ji}^{\alpha\alpha}$$

- Full equation of motion for classical spin-components:

$$\frac{d}{dt}x_m = z_m\Big(2b_m^y + 2\sum_j V_{mj}^{yy}y_j\Big) + y_m\Big(-2b_m^z - 2\sum_j V_{mj}^{zz}z_j\Big)$$

$$\frac{d}{dt}y_m = z_m\Big(-2b_m^x - 2\sum_j V_{mj}^{xx}x_j\Big) + x_m\Big(2b_m^z + 2\sum_j V_{mj}^{zz}z_j\Big)$$

$$\frac{d}{dt}z_m = x_m\Big(-2b_m^y - 2\sum_j V_{mj}^{yy}y_j\Big) + y_m\Big(2b_m^x + 2\sum_j V_{mj}^{xx}x_j\Big)$$

precession due to external fields and mean-fields of other spins

- **Remark:** Here all was derived for Pauli matrices, for spin operators, in the literature one often finds equations for spin-components:

$$s_m^x = \frac{1}{2}x_m$$

warning … many funny factors of two!

Mean-field approach to spin-models

- Full equation of motion for classical spin-components:

$$\frac{d}{dt}x_m = z_m\big(2b_m^y + 2\sum_j V_{mj}^{yy}y_j\big) + y_m\big(-2b_m^z - 2\sum_j V_{mj}^{zz}z_j\big)$$
$$\frac{d}{dt}y_m = z_m\big(-2b_m^x - 2\sum_j V_{mj}^{xx}x_j\big) + x_m\big(2b_m^z + 2\sum_j V_{mj}^{zz}z_j\big)$$
$$\frac{d}{dt}z_m = x_m\big(-2b_m^y - 2\sum_j V_{mj}^{yy}y_j\big) + y_m\big(2b_m^x + 2\sum_j V_{mj}^{xx}x_j\big)$$

precession due to external fields and mean-fields of other spins

- **Equations** for our transverse Ising model: $\hat{H} = \frac{1}{2}\sum_{i\neq j} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z + h_x\sum_i \hat{\sigma}_i^x$

$$V_{ij}^{xx} = V_{ij}^{yy} = 0 \qquad b_m^y = b_m^z = 0 \qquad V_{ij}^{zz} = J_{ij} = \frac{J}{|i-j|^\alpha} \qquad b_m^x = h_x$$

- Then:

$$\frac{d}{dt}x_m = -y_m\big(2\sum_j V_{mj}^{zz}z_j\big)$$
$$\frac{d}{dt}y_m = x_m\big(2\sum_j V_{mj}^{zz}z_j\big) - 2z_m h_x$$
$$\frac{d}{dt}z_m = 2y_m h_x$$

zz - interactions -> precession in x-y plane

field || x -> precession in y-z plane

Mean-field approach to spin-models

- **Mean-field equations** for transverse Ising model:

$$\frac{d}{dt}x_m = -y_m\big(2\sum_j V_{mj}^{zz} z_j\big)$$
$$\frac{d}{dt}y_m = x_m\big(2\sum_j V_{mj}^{zz} z_j\big) - 2z_m h_x$$
$$\frac{d}{dt}z_m = 2y_m h_x$$

$$x_i(t) \equiv \langle \hat{\sigma}_i^x(t)\rangle$$
$$y_i(t) \equiv \langle \hat{\sigma}_i^y(t)\rangle$$
$$z_i(t) \equiv \langle \hat{\sigma}_i^z(t)\rangle$$

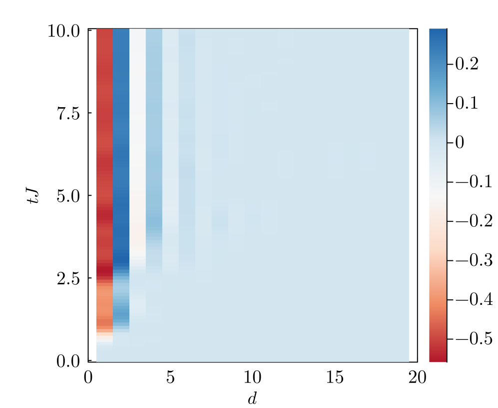
- We now reduced the problem to a set of only $3N$ coupled equations!

Question: Idea to make use of even less memory?

- … but we pay a price: **I. We made a strong approximation, II. The equations are now non-linear**

- For example, the two-point correlations will now be trivial:

$$C_{i,j} = \langle \hat{\sigma}_i^z \hat{\sigma}_j^z\rangle - \langle \hat{\sigma}_i^z\rangle\langle \hat{\sigma}_j^z\rangle = \langle \hat{\sigma}_i^z\rangle\langle \hat{\sigma}_j^z\rangle - \langle \hat{\sigma}_i^z\rangle\langle \hat{\sigma}_j^z\rangle = 0$$

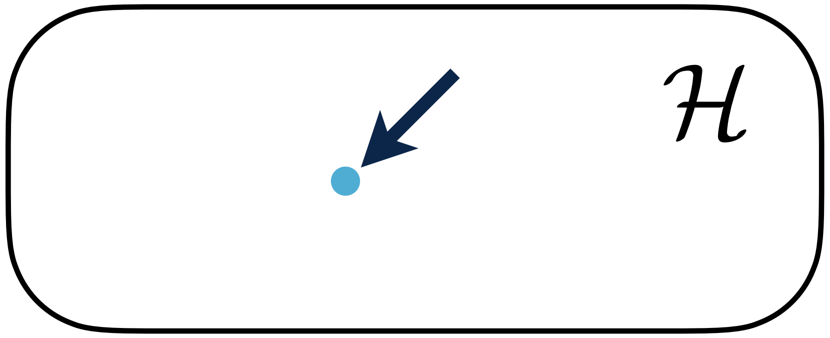


$= 0$

This time

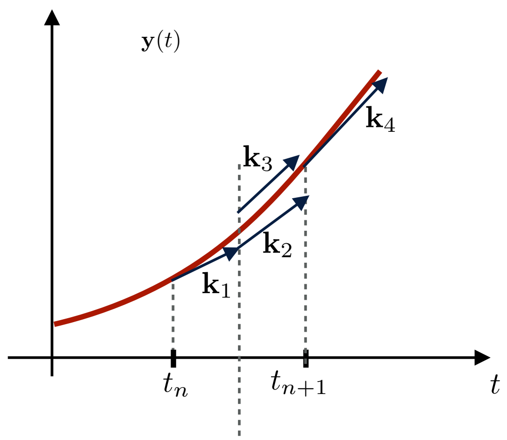
- **Part 1:** Mean-field approach to spin-models

$$|\psi\rangle \approx |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_N\rangle$$

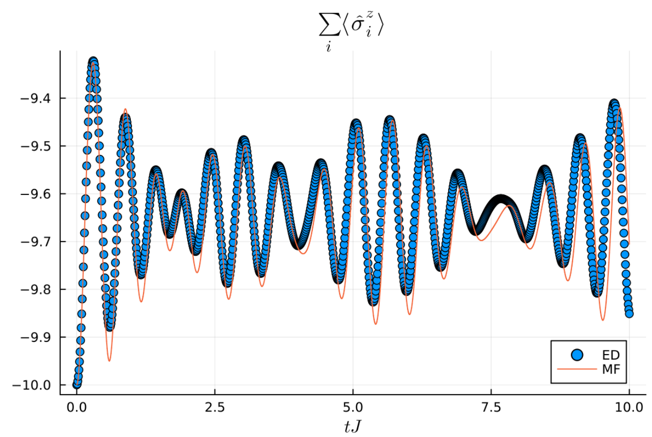


- **Part 2:** Runge-Kutta (RK) time-evolution methods: A swiss army knife

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



- **Tutorial:** A mean-field simulation of the transverse Ising model



- **Part 3:** Mean-field dynamics for bosonic systems (the Gross-Pitaevskii equation)

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

Runge-Kutta Methods

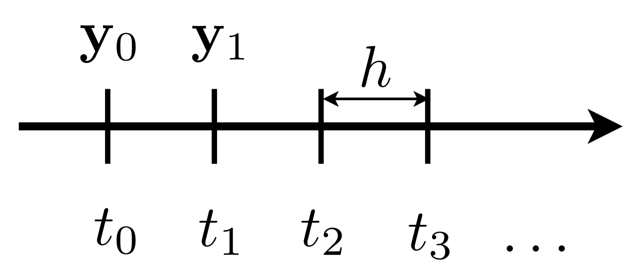
- A general class of standard methods for initial value problems ("Swiss army knife")

$$\boxed{\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)}$$

$\mathbf{y}(\dots)$ *"exact"*

$\mathbf{y}_n$ *"numerical approximation"*

- Time-discretization:



- Note: This includes linear Schrödinger equation, but also non-linear problems

Schrödinger equation

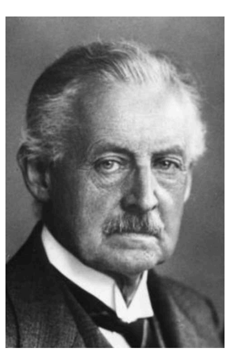
$$\frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle$$

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

mean-field equations

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

$$\mathbf{y} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ y_1 \\ y_2 \\ \vdots \\ y_N \\ z_1 \\ z_2 \\ \vdots \\ z_N \end{pmatrix}$$



Carl David Tolmé Runge
(1856-1927)

Martin Kutta
(1867–1944)

Runge-Kutta Methods: 1st order - explicit Euler

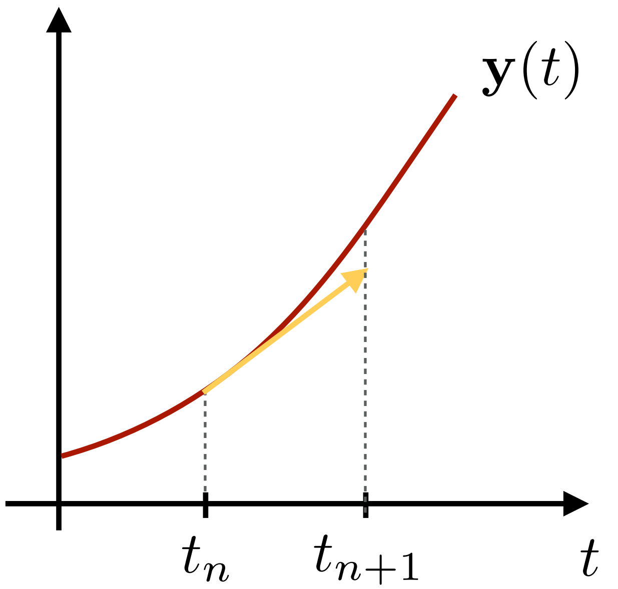
$$\boxed{\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)}$$

- Let's find a method from Taylor expansion of $\mathbf{y}(t)$

$$\mathbf{y}(t_n + h) = \mathbf{y}(t_n) + h\dot{\mathbf{y}}(t_n) + \frac{h^2}{2}\ddot{\mathbf{y}}(t_n) + \cdots = \mathbf{y}(t_n) + hf(t_n, \mathbf{y}_n) + \mathcal{O}(h^2)$$

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + hf(t_n, \mathbf{y}_n)$$

"explicit Euler method"

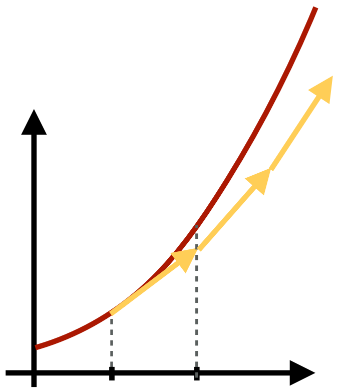


- Not a good method for several reasons

Error is large $\mathcal{O}(h^2)$

…need tiny h

Solution is often not stable!



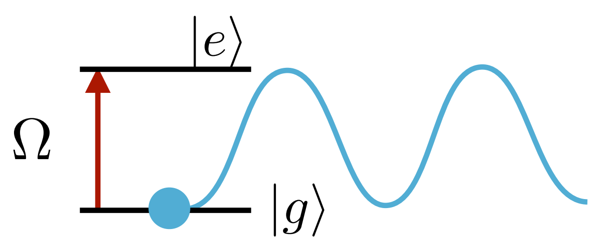
"error grows in same direction"

Runge-Kutta Methods: 1st order

$$\boxed{\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)}$$

- … very simple example (Rabi oscillations)

$$\hat{H} = \begin{pmatrix} 0 & \Omega \\ \Omega & 0 \end{pmatrix} \qquad \frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle \qquad |\psi_0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{array}{c} |g\rangle \\ |e\rangle \end{array}$$



Compute:

$$n_e(t) = |\,\langle\psi(t)|e\rangle\,|^2$$

Exact:

$$n_e(t) = \sin^2(t\Omega)$$

$\Omega \equiv 1$

```
h = 0.05
steps = 200

psi = [1;0]
ne = zeros(steps+1)
ne[1] = abs(psi[2])^2
for tt = 1:steps
    psi = rk1(H, psi, h)
    ne[tt+1] = abs(psi[2])^2
end
```

- Explicit Euler method:

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + hf(t_n, \mathbf{y}_n)$$

“explicit Euler method”

```
function rk1(H, y, h)
    y += h .* (-1im .* H * y)
    return y
end
```

Runge-Kutta Methods: 1st order

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

- Explicit Euler method: $\mathbf{y}_{n+1} \approx \mathbf{y}_n + h f(t_n, \mathbf{y}_n)$

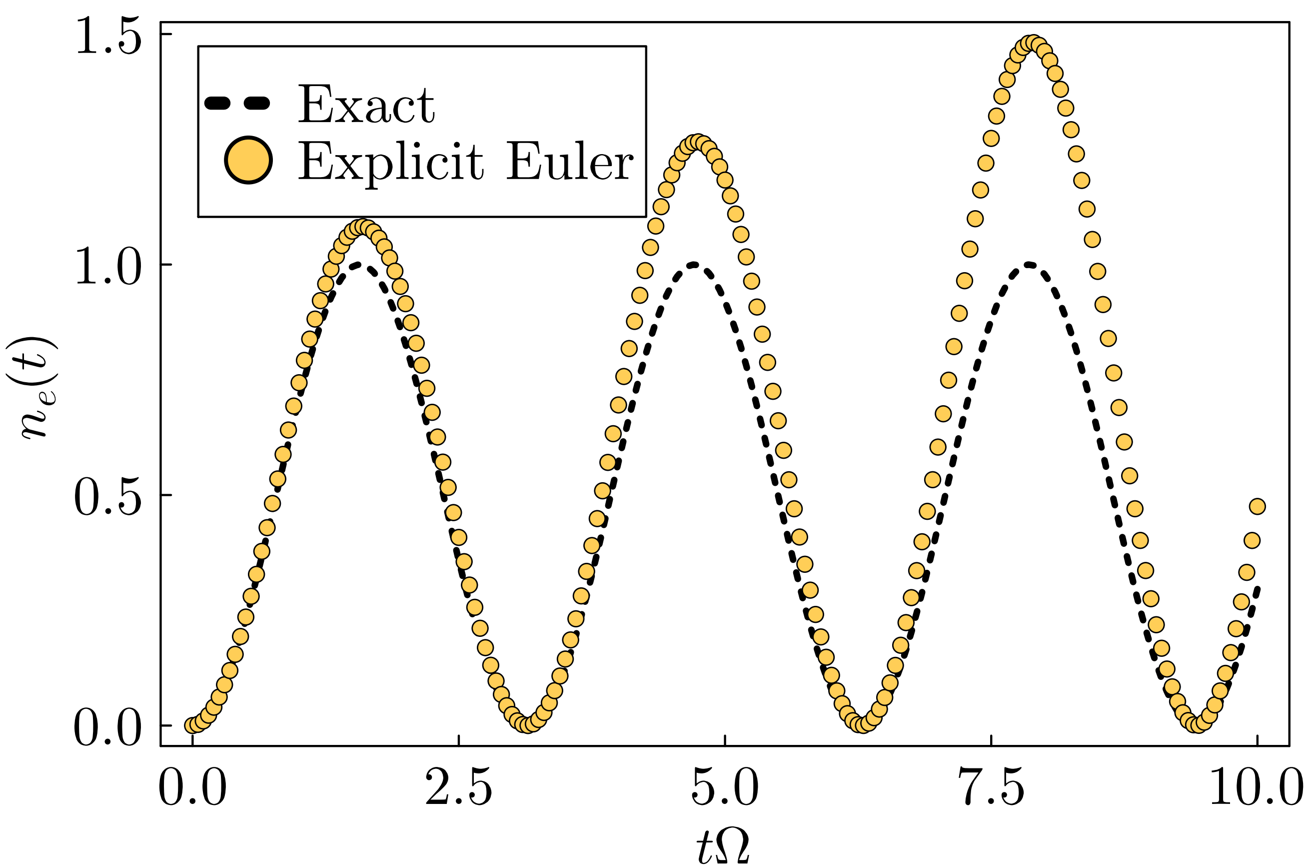
"explicit Euler method"

```
function rk1(H, y, h)
    y += h .* (-1im .* H * y)
    return y
end
```

- … very simple example (Rabi oscillations)

$$\hat{H} = \begin{pmatrix} 0 & \Omega \\ \Omega & 0 \end{pmatrix} \qquad \frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle \qquad |\psi_0\rangle = \begin{array}{cc} \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{array}{c} |g\rangle \\ |e\rangle \end{array} \end{array}$$

$$n_e(t) = |\langle\psi(t)|e\rangle|^2$$



Fundamental problem:
Norm keeps increasing!

$$|\psi_{n+1}\rangle = |\psi_n\rangle - \mathrm{i}h\hat{H}\,|\psi_n\rangle$$

$$\langle\psi_{n+1}|\psi_{n+1}\rangle = \langle\psi_n|\psi_n\rangle + h^2\,\langle\psi_n|\,\hat{H}^2\,|\psi_n\rangle$$

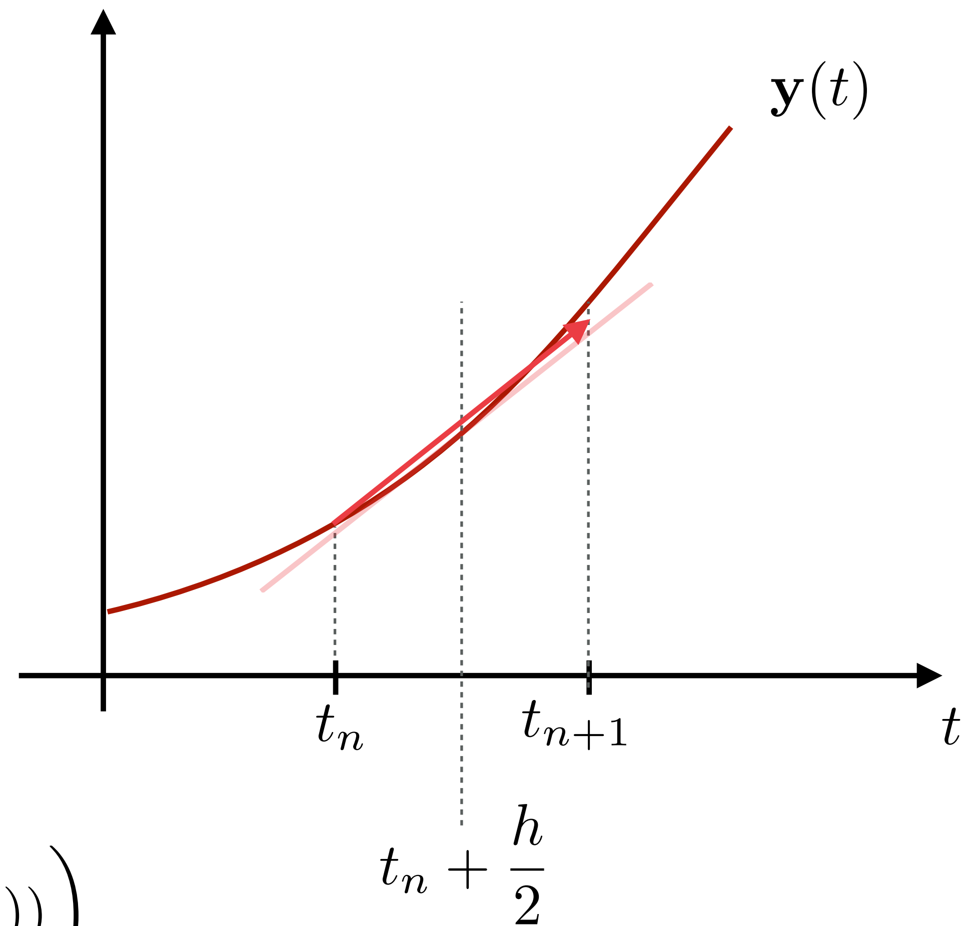
Runge-Kutta Methods: 2nd order - Midpoint

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

- Let's find a better method:

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + hf\left(t_n + \frac{1}{2}, \frac{1}{2}(\mathbf{y}_n + \mathbf{y}_{n+1})\right)$$

"take slope at middle point"



- Then, in exact Taylor expansion, the error is

$$\epsilon_{n+1} \equiv \mathbf{y}(t_n + h) - \mathbf{y}(t_n) - hf\left(t_n + \frac{h}{2}, \frac{1}{2}(\mathbf{y}(t_n) + \mathbf{y}(t_{n+1}))\right)$$
$$= \cdots = 0 + \mathcal{O}(h^3)$$

(exercise)

- This is called "implicit midpoint method"

Implicit, meaning: The right hand-side has already the solution at n+1, so one generally needs to resolve the equation for the n+1 value or use some iteration.

Runge-Kutta Methods: 2nd order - Midpoint

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + hf\left(t_n + \frac{1}{2}, \frac{1}{2}(\mathbf{y}_n + \mathbf{y}_{n+1})\right)$$

"implicit midpoint method"

- Zero order iteration gives:

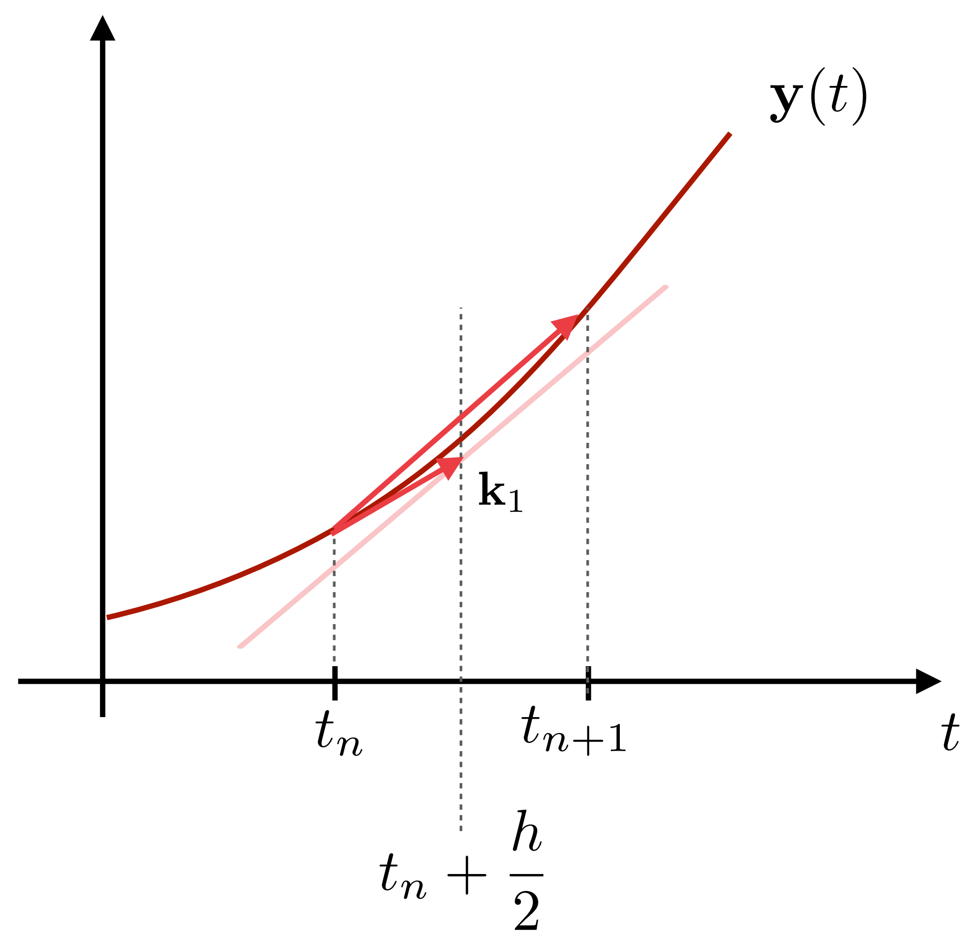
$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + hf\left(t_n + \frac{1}{2}, \mathbf{y}_n\right) \qquad \mathbf{k}_1 \equiv \frac{\mathbf{y}_n + \mathbf{y}_{n+1}}{2} \approx \mathbf{y}_n + \frac{h}{2} f\left(t_n + \frac{1}{2}, \mathbf{y}_n\right)$$

Explicit Euler estimate for mid-point

$$\mathbf{k}_1 = \mathbf{y}_n + \frac{h}{2} f(t_n + \mathbf{y}_n)$$

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + hf(t_n + \frac{h}{2}, \mathbf{k}_1)$$

"explicit midpoint method"



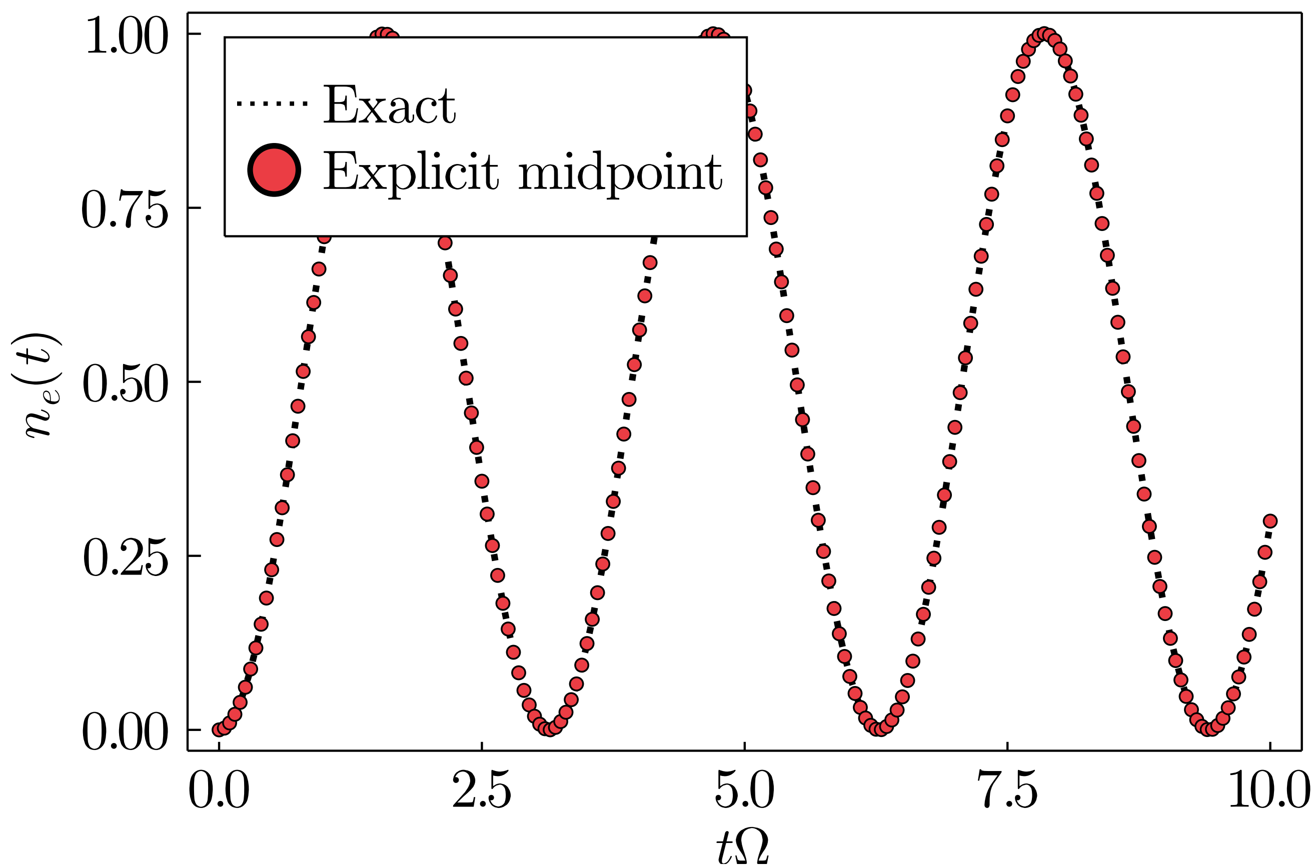
Runge-Kutta Methods: 2nd order - Midpoint

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

- … very simple example (Rabi oscillations)

$$\hat{H} = \begin{pmatrix} 0 & \Omega \\ \Omega & 0 \end{pmatrix} \qquad \frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle \qquad |\psi_0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{array}{l} |g\rangle \\ |e\rangle \end{array}$$

$$n_e(t) = |\langle\psi(t)|e\rangle|^2$$



$$\mathbf{k}_1 = \mathbf{y}_n + \frac{h}{2} f(t_n + \mathbf{y}_n)$$

$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + h f(t_n + \frac{h}{2}, \mathbf{k}_1)$$

"explicit midpoint method"

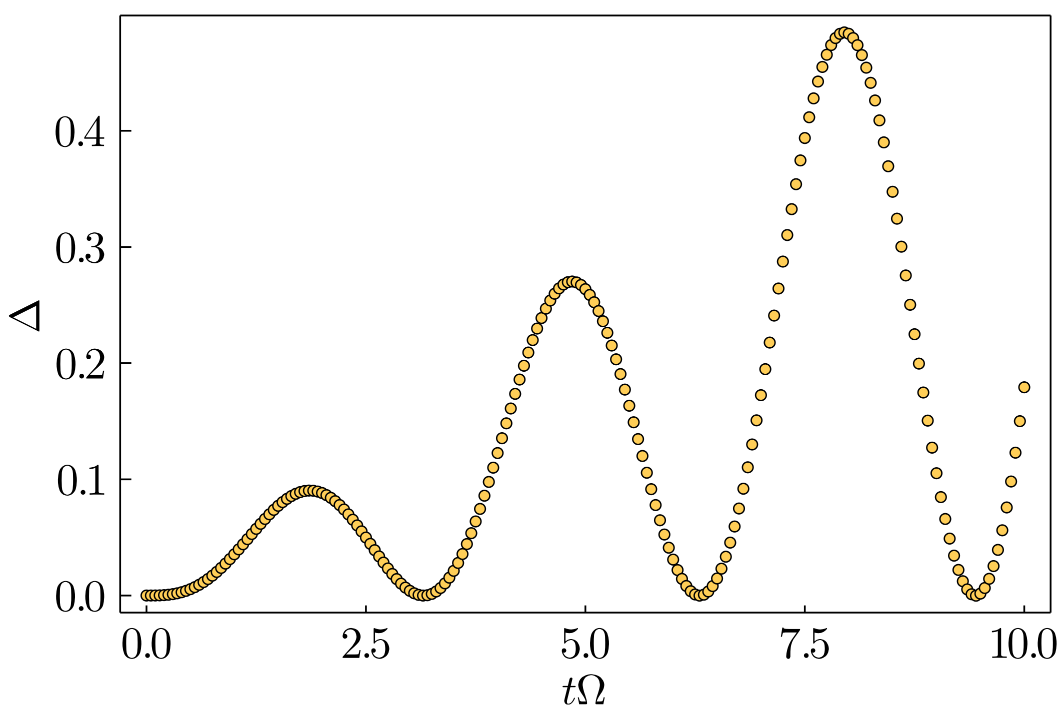
```
function rk2e(H, y, h)
    k1 = y .+ (h/2) .* (-1im .* H * y)
    y += h .* (-1im .* H * k1)
    return y
end
```

Runge-Kutta Methods: 2nd order - Midpoint

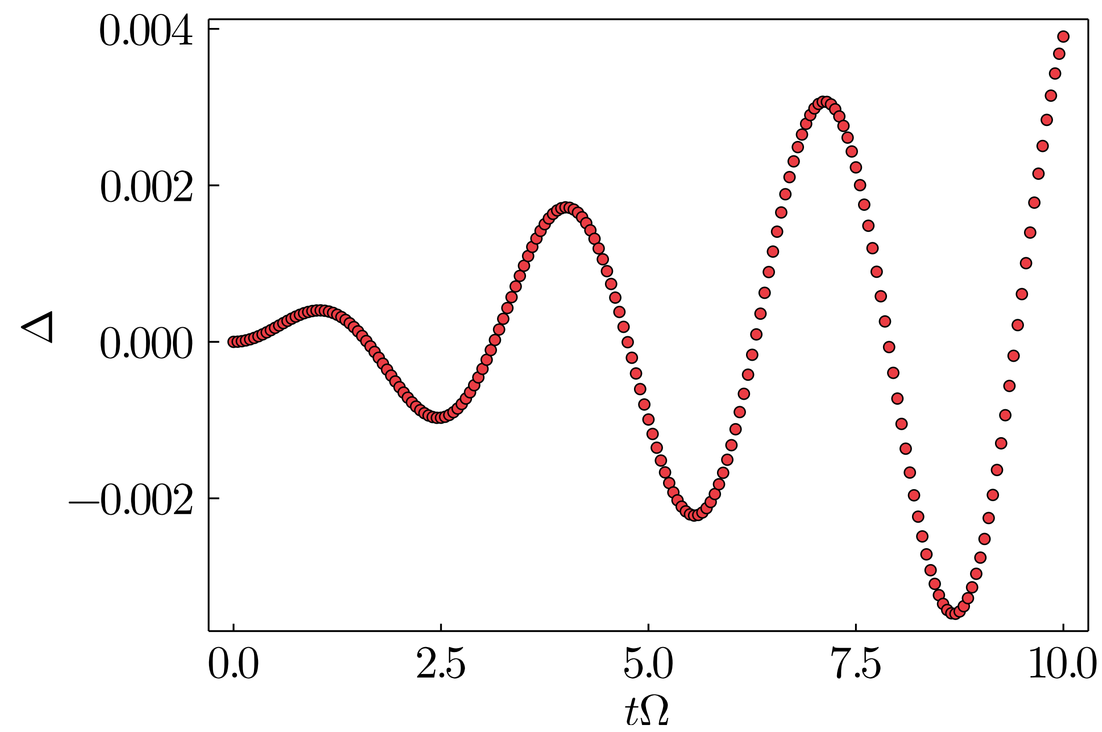
$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

- Comparisons $\Delta \equiv n_e(t) - \sin^2(t\Omega)$ $\qquad h\Omega = 0.05$

"explicit Euler"



"explicit midpoint"



Stable!

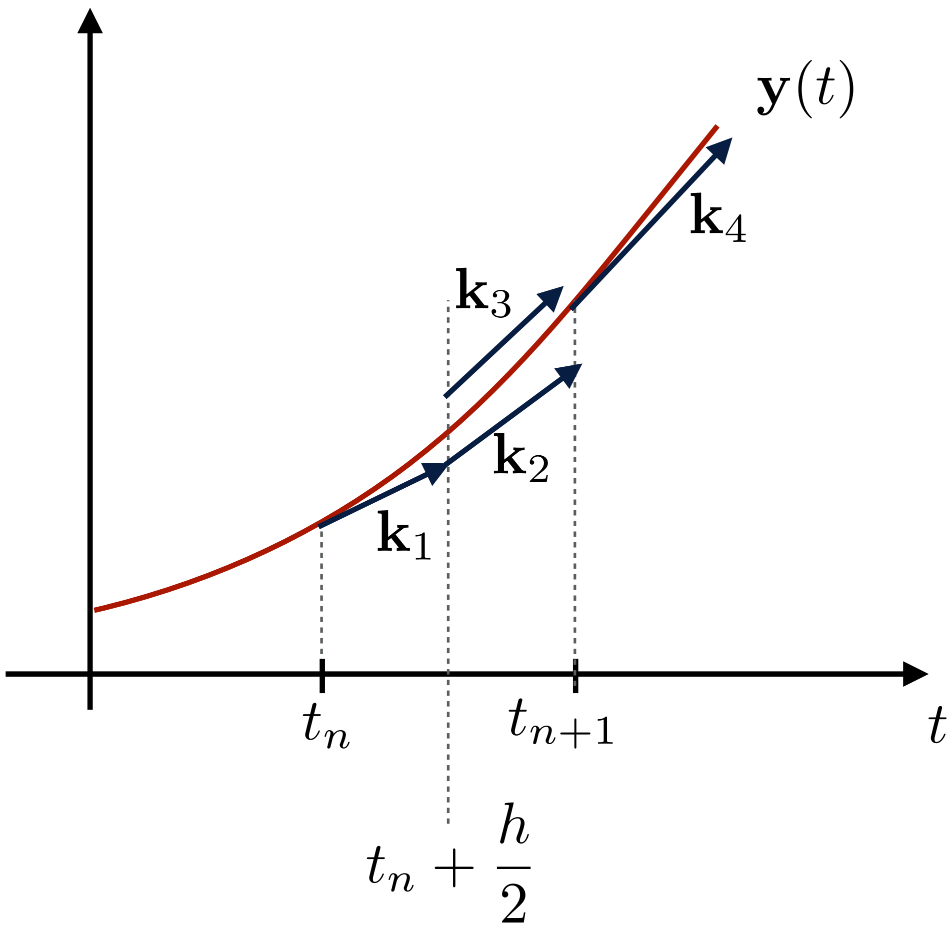
Runge-Kutta Methods: 4th order

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

- In practice often the most convenient method

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

"4th order Runge-Kutta"



- **Remarks:**

  - *Local error* $\epsilon_{n+1} = \mathcal{O}(h^5)$
  - *Note: n-th order = n function evaluations … higher order pays off!*
  - *In practice n=4 is convenient: e.g. 100 steps for plots, typical timescales ~10, time-step ~0.1 ideal*

Runge-Kutta Methods: 4th order

$$\dot{\mathbf{y}}(t) = f(t, \mathbf{y}(t)) \qquad \mathbf{y}(t_n) = \mathbf{y}_n \qquad \textit{Find:} \quad \mathbf{y}_{n+1} \approx \mathbf{y}(t_n + h)$$

$$\mathbf{k}_1 = f(t_n, \mathbf{y}_n)$$

$$\mathbf{k}_2 = f\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1\right)$$

$$\mathbf{k}_3 = f\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2\right)$$

$$\mathbf{k}_4 = f\left(t_n + h, \mathbf{y}_n + h\mathbf{k}_3\right)$$

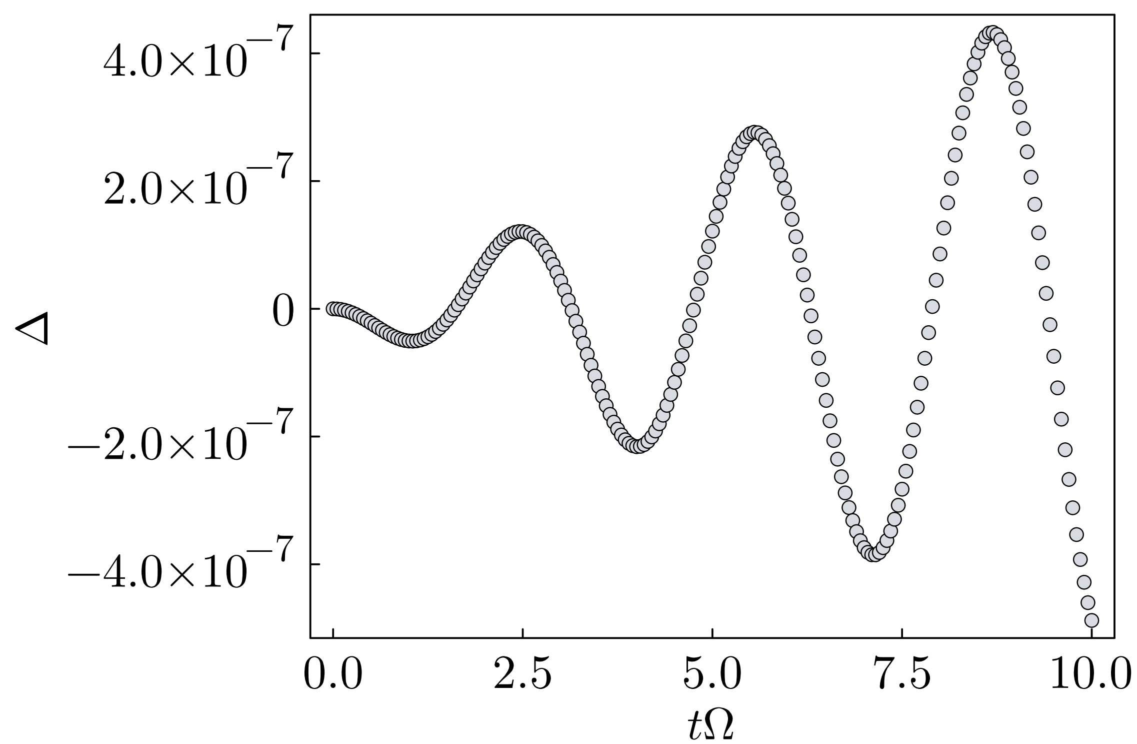
$$\mathbf{y}_{n+1} \approx \mathbf{y}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

"4th order Runge-Kutta"

```
function rk4(H, y, h)
    h2 = h/2
    imH = -1im .* H
    k1 = imH * y
    k2 = imH * (y .+ h2 .* k1)
    k3 = imH * (y .+ h2 .* k2)
    k4 = imH * (y .+ h .* k3)
    y += (h/6) .* (k1 .+ 2 .* k2 .+ 2 .* k3 .+ k4)
    return y
end
```

- … very simple example (Rabi oscillations)

$$\Delta \equiv n_e(t) - \sin^2(t\Omega)$$

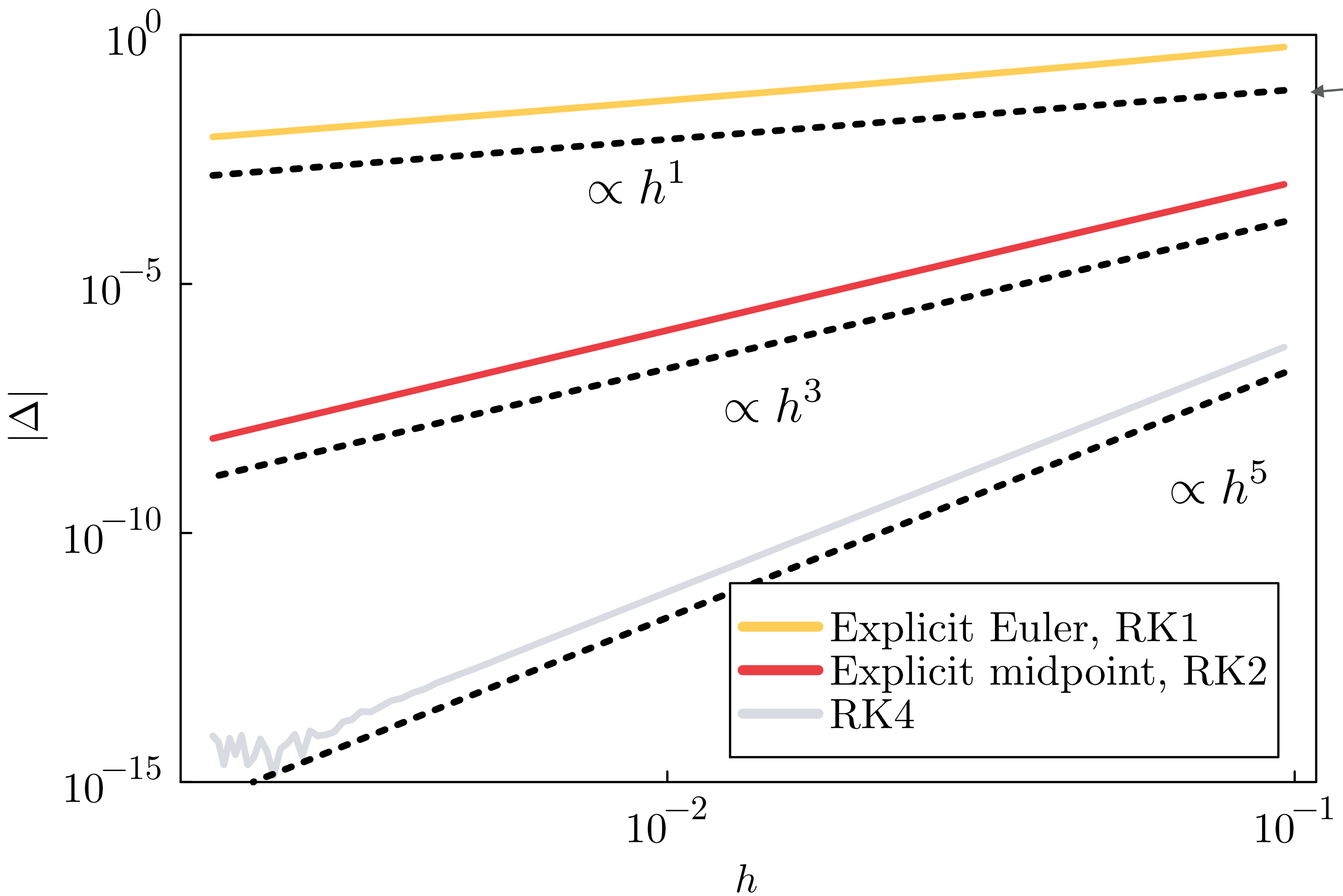


Runge-Kutta Methods: Sanity checks

- … very simple example (Rabi oscillations) $\hat{H} = \begin{pmatrix} 0 & \Omega \\ \Omega & 0 \end{pmatrix}$ $\frac{d}{dt}|\psi\rangle = -\mathrm{i}\hat{H}|\psi\rangle$ $|\psi_0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{matrix} |g\rangle \\ |e\rangle \end{matrix}$

$$\Delta \equiv n_e(t) - \sin^2(t\Omega)$$

- Error at time fixed time, compare methods: $t\Omega = 3\frac{\pi}{2}$ $\sin^2(t\Omega) = 1$

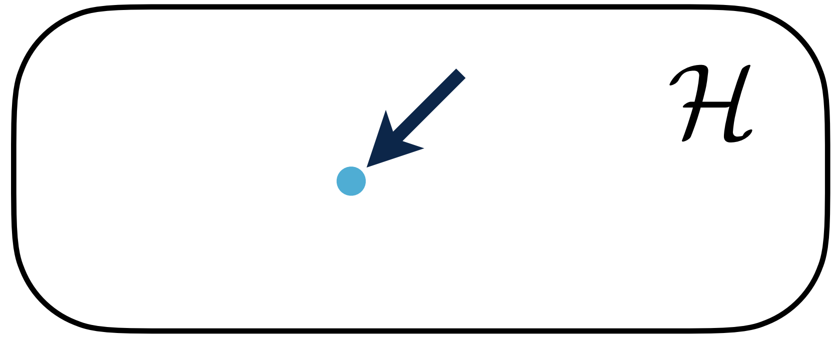


why not? $\propto h^2$

This time

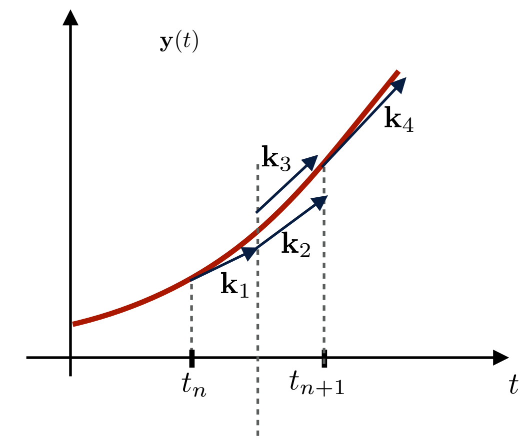
- **Part 1:** Mean-field approach to spin-models

$$|\psi\rangle \approx |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_N\rangle$$

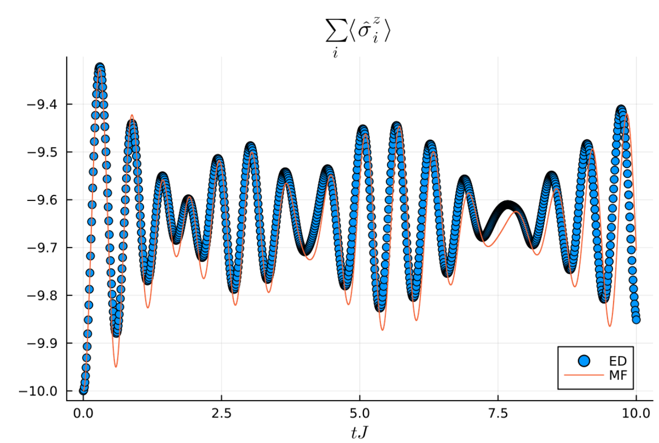


- **Part 2:** Runge-Kutta (RK) time-evolution methods: A swiss army knife

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



- **Tutorial:** A mean-field simulation of the transverse Ising model



- **Part 3:** Mean-field dynamics for bosonic systems (the Gross-Pitaevskii equation)

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

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

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

$$\frac{d}{dt}x_m = -y_m\big(2\sum_j V_{mj}^{zz}z_j\big)$$
$$\frac{d}{dt}y_m = x_m\big(2\sum_j V_{mj}^{zz}z_j\big) - 2z_m h_x$$
$$\frac{d}{dt}z_m = 2y_m h_x$$

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

Initial state: $|\psi_0\rangle = |\downarrow\downarrow\ldots\downarrow\rangle$

Observable: spin-z magnetization $m_z = \sum_i \langle\hat{\sigma}_i^z\rangle$

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

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

$$\frac{d}{dt}x_m = -y_m\big(2\sum_j V_{mj}^{zz} z_j\big)$$
$$\frac{d}{dt}y_m = x_m\big(2\sum_j V_{mj}^{zz} z_j\big) - 2z_m h_x$$
$$\frac{d}{dt}z_m = 2y_m h_x$$

$$\mathbf{y} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ y_1 \\ y_2 \\ \vdots \\ y_N \\ z_1 \\ z_2 \\ \vdots \\ z_N \end{pmatrix} \qquad \mathbf{y}(t=0) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}$$

Initial state: $|\psi_0\rangle = |\downarrow\downarrow\ldots\downarrow\rangle$

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

$$\frac{d}{dt}x_m = -y_m\left(2\sum_j V^{zz}_{mj} z_j\right)$$
$$\frac{d}{dt}y_m = x_m\left(2\sum_j V^{zz}_{mj} z_j\right) - 2z_m h_x$$
$$\frac{d}{dt}z_m = 2y_m h_x$$

Field, following from matrix multiplication

@views for pointer arithmetics

```
function f_tising(Y, Vij, hx, N)

    xran = 1:N
    yran = (N+1):(2*N)
    zran = (2*N+1):(3*N)

    dY = Vector{Float64}(undef, 3*N)

    @views mfs = Vij * Y[zran]  # vector of mean-fields
    @views dY[xran] = - mfs .* Y[yran]
    @views dY[yran] = mfs .* Y[xran] - hx .* Y[zran]
    @views dY[zran] = hx .* Y[yran]

    return 2 .* dY

end
```

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation
2. Construct a function for a RK4 time-step
3. Compute and plot observable dynamics

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

$$\frac{d}{dt}x_m = -y_m\big(2\sum_j V_{mj}^{zz} z_j\big)$$
$$\frac{d}{dt}y_m = x_m\big(2\sum_j V_{mj}^{zz} z_j\big) - 2z_m h_x$$
$$\frac{d}{dt}z_m = 2y_m h_x$$

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

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

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

```
function rk4_tising(Y, Vij, hx, dt, N)

    f(Y) = f_tising(Y, Vij, hx, N)

    dt2 = dt/2
    k1 = f(Y)
    k2 = f(Y .+ dt2 .* k1)
    k3 = f(Y .+ dt2 .* k2)
    k4 = f(Y .+ dt .* k3)
    Y += (dt/6) .* (k1 .+ 2 .* k2 .+ 2 .* k3 .+ k4)

    return Y
end
```

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

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

Initial state: $|\psi_0\rangle = |\downarrow\downarrow\ldots\downarrow\rangle$

Observable: spin-z magnetization $m_z = \sum_i \langle\hat{\sigma}_i^z\rangle$

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

```
function main()


    N = 10
    J = 1
    hx = 1
    alpha = 3

    dt = 0.1
    tran = 0:dt:10
    steps = length(tran)

    # Build coupling matrix
    Jij = Matrix{Float64}(undef, N, N)
    for ii = 1:N
        Jij[ii,ii] = 0.0
        for jj = (ii+1):N
            Jij[ii,jj] = J/(jj-ii)^alpha
            Jij[jj,ii] = Jij[ii,jj]
        end
    end
```

```
# exact simulation
Sz_ed = sparse_ti_simulation(N, 1, alpha, hx, dt, steps

plt = scatter(tran, sum(Sz_ed; dims=2), labels="ED")
xlabel!(L"tJ")
title!(L" \sum_i \langle \hat \sigma^z_i \rangle")
display(plt)
return nothing
```

Play with field, analogy of optical Bloch equations (rest of system acts as bath)
… weakly "driven system" more interesting.

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

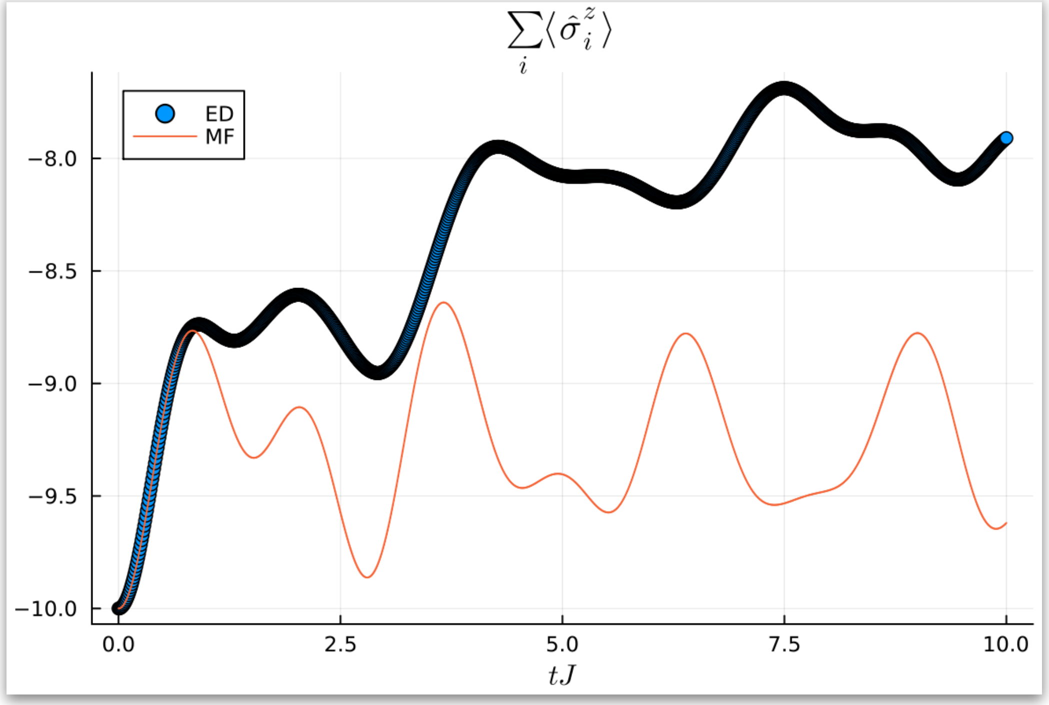
2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

```
    # initial state
    Y = [zeros(2*N); -ones(N)]

    Sz_mf = Matrix{Float64}(undef, steps, N)
    for tt = 1:steps
        Sz_mf[tt, :] .= Y[(2 * N + 1):end] # evaluate
        Y = rk4_tising(Y, Jij, hx, dt, N) # evolve
    end


    plt = plot()
    scatter!(tran, sum(Sz_ed; dims=2), labels="ED")
    plot!(tran, sum(Sz_mf; dims=2), labels="MF")
    xlabel!(L"tJ")
    title!(L" \sum_i \langle \hat \sigma^z_i \rangle")
    display(plt)

    return nothing

end
```

Play with field, low fields -> MF exact!

$$\alpha = 3, h_x = 0.5J$$



Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

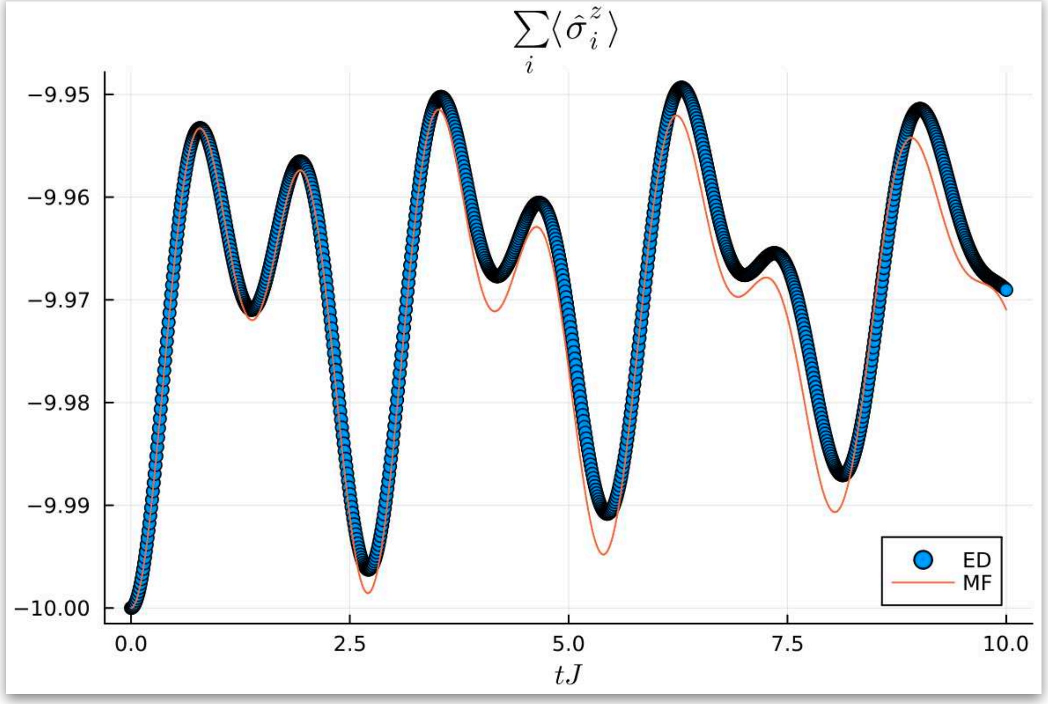
2. Construct a function for a RK4 time-step

3. Compute and plot observable dynamics

```
    # initial state
    Y = [zeros(2*N); -ones(N)]

    Sz_mf = Matrix{Float64}(undef, steps, N)
    for tt = 1:steps
        Sz_mf[tt, :] .= Y[(2 * N + 1):end] # evaluate
        Y = rk4_tising(Y, Jij, hx, dt, N) # evolve
    end


    plt = plot()
    scatter!(tran, sum(Sz_ed; dims=2), labels="ED")
    plot!(tran, sum(Sz_mf; dims=2), labels="MF")
    xlabel!(L"tJ")
    title!(L" \sum_i \langle \hat \sigma^z_i \rangle")
    display(plt)

    return nothing

end
```

Play with field, low fields -> MF exact!

$$\alpha = 3, h_x = 0.1J$$



Why? Low excitation, interactions not very important!

Question: Other limit, where MF becomes good?

Tutorial: A mean-field simulation for the transverse Ising model

1. Design a state vector and construct a function for the RHS of the non-linear equation

2. Construct a function for a RK4 time-step

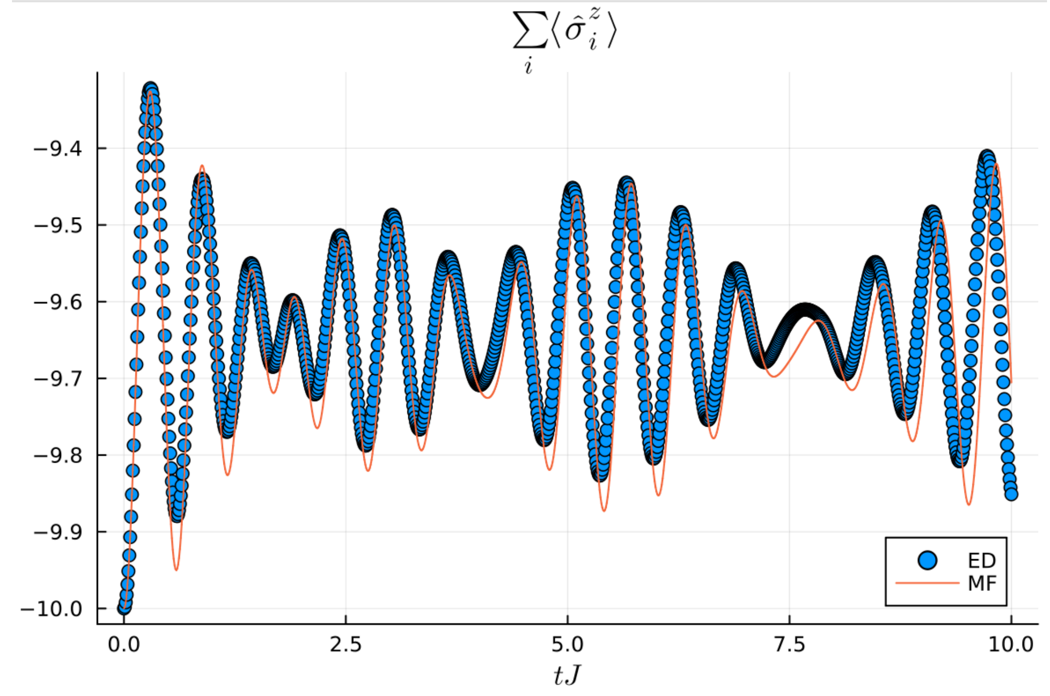
3. Compute and plot observable dynamics

```
    # initial state
    Y = [zeros(2*N); -ones(N)]

    Sz_mf = Matrix{Float64}(undef, steps, N)
    for tt = 1:steps
        Sz_mf[tt, :] .= Y[(2 * N + 1):end] # evaluate
        Y = rk4_tising(Y, Jij, hx, dt, N) # evolve
    end


    plt = plot()
    scatter!(tran, sum(Sz_ed; dims=2), labels="ED")
    plot!(tran, sum(Sz_mf; dims=2), labels="MF")
    xlabel!(L"tJ")
    title!(L" \sum_i \langle \hat \sigma^z_i \rangle")
    display(plt)

    return nothing

end
```

Question: Other limit, where MF becomes good?

Strong drive, of course, but also … alpha small!

$$\alpha = 0.5, h_x = J$$

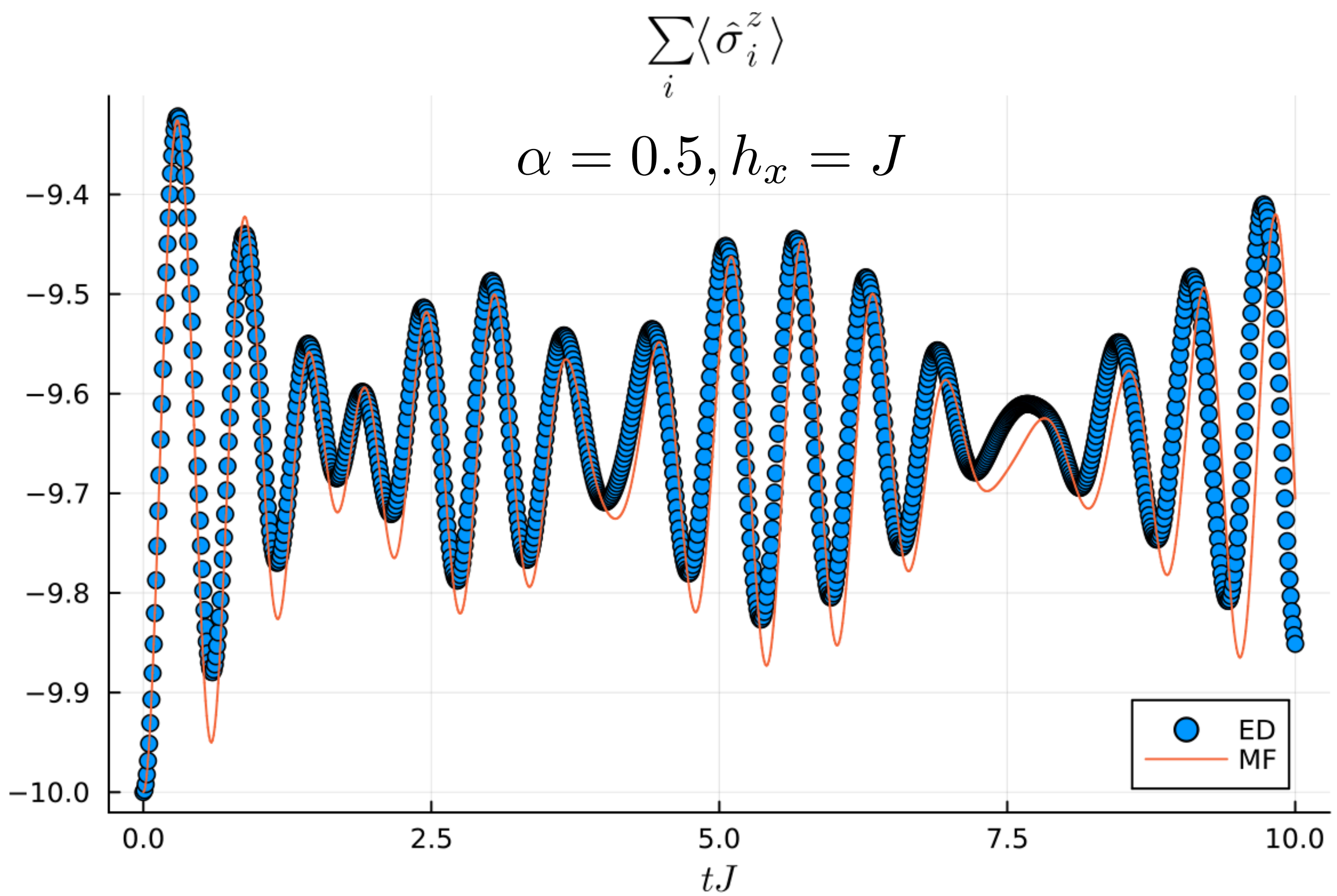


Tutorial: A mean-field simulation for the transverse Ising model

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

Initial state: $|\psi_0\rangle = |\downarrow\downarrow\ldots\downarrow\rangle$

Observable: spin-z magnetization $m_z = \sum_i \langle\hat{\sigma}_i^z\rangle$



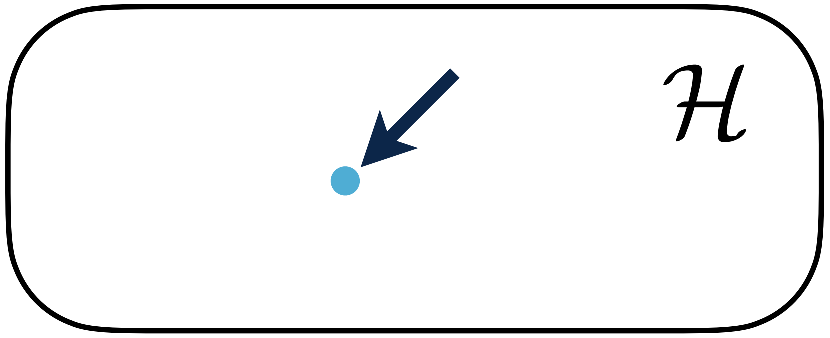
- **Note:** The mean-field approximation is quite good in low-excitation regimes, and for "high connectivity"?

Well-known: Mean-field works good in higher dimension, worst-case: 1D with nearest-neighbor couplings

This time

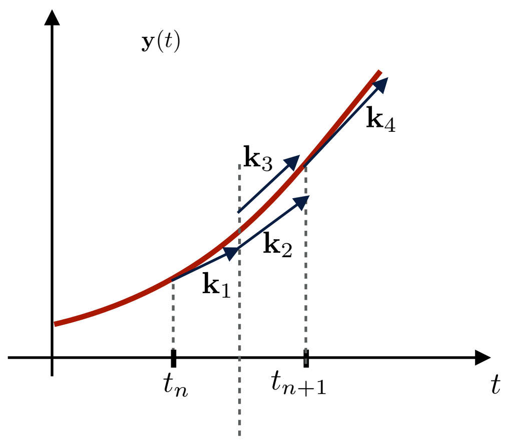
- **Part 1:** Mean-field approach to spin-models

$$|\psi\rangle \approx |\phi_1\rangle \otimes |\phi_2\rangle \otimes \cdots \otimes |\phi_N\rangle$$

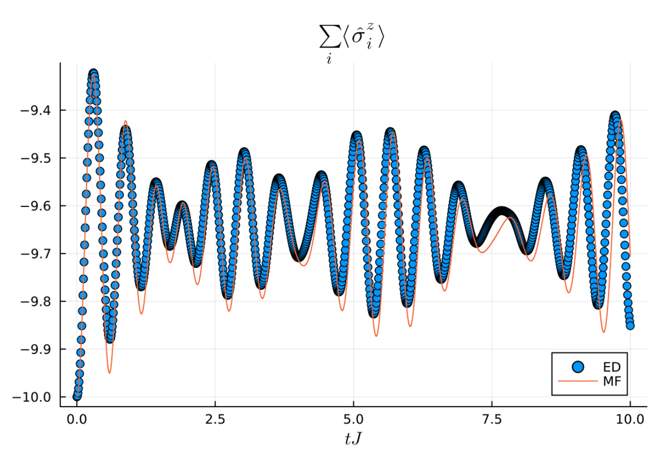


- **Part 2:** Runge-Kutta (RK) time-evolution methods:
  A swiss army knife

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



- **Tutorial:** A mean-field implementation for the transverse Ising model



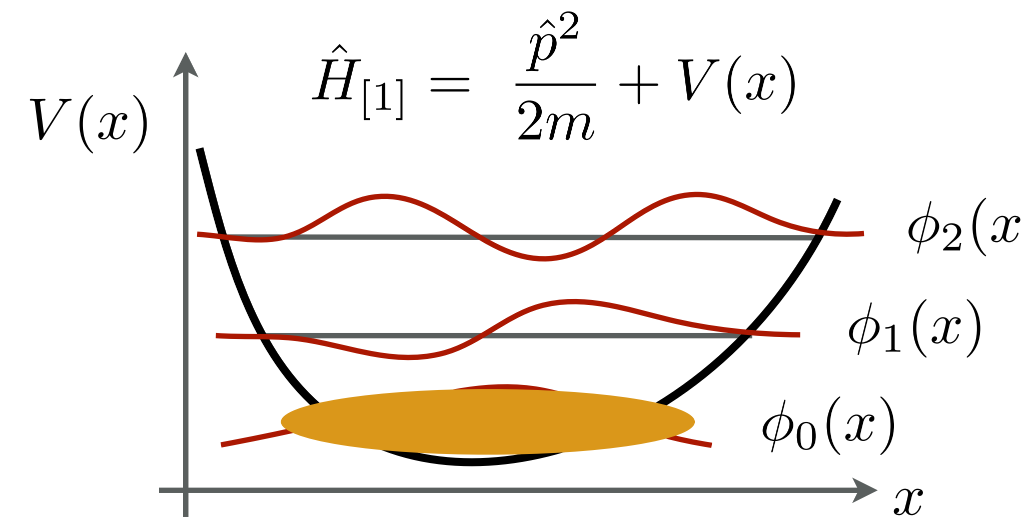
- **Part 3:** Mean-field dynamics for bosonic systems (the Gross-Pitaevskii equation)

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

One-pager: GP equation

- Dilute bosonic gas (N particles) with contact interactions:

Bosonic field operator: $$\boxed{\frac{d}{dt}\hat{\psi}_x = -\mathrm{i}\left(\hat{H}_{[1]} + g\hat{\psi}_x^\dagger\hat{\psi}_x\right)\hat{\psi}_x}$$



Almost all particles are in the lowest trapped state

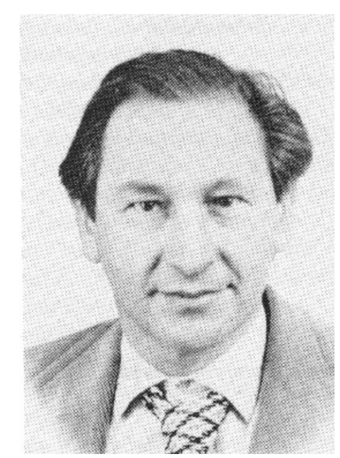
- Classical mean-field approximation for the quantum field:

$$\hat{\psi}_x = \sum_n \phi_n(x)\hat{a}_n = \phi_0(x)\hat{a}_0 + \cancel{\sum_{n>0}}\hat{a}_n \approx \sqrt{N}\phi_0(x) \equiv \psi(x)$$

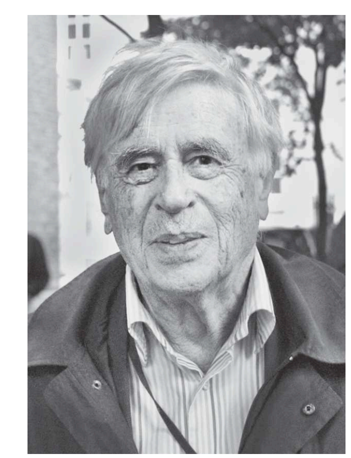
Bosonic field operators for trapped states

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

Time-dependent Gross-Pitaevskii equation!



Eugene P. Gross (1926-1991)



Lev Pitaevskii (1933-2022)

- A **non-linear** equation for a classical field $\psi(x,t)$ describing a condensate of many particles

- This is equivalent to saying the many-body system remains in a **product state:**

$$\phi(x_1, x_2, \ldots, x_N) = \bigotimes_{i=1}^{N} \phi_0(x_i)$$

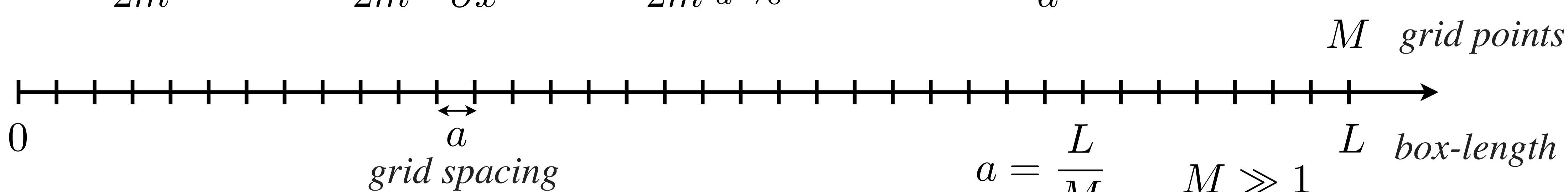
(alternative derivation)

GP equation - Numerical considerations

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

- For a continuous space, we need to introduce a discretization

$$\frac{\hat{p}^2}{2m}\psi(x) = -\frac{1}{2m}\frac{\partial^2\psi(x)}{\partial x^2} = -\frac{1}{2m}\lim_{a\to 0}\frac{\psi(x+a) - 2\psi(x) + \psi(x-h)}{a^2}$$



$$a = \frac{L}{M} \qquad M \gg 1$$

- Then the matrix of the kinetic energy is:

$$\frac{1}{2ma^2}\begin{pmatrix} -1 & 0 & -1 & & & & \\ & -1 & 0 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 0 & -1 & \\ & & & & -1 & 0 & -1 \end{pmatrix} + \text{const.} \qquad \text{… acting on …} \qquad \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{M-1} \\ \psi_M \end{pmatrix}$$

- **Note:** Kinetic energy is nearest-neighbor hopping on a grid with amplitude $J = \frac{1}{2ma^2}$

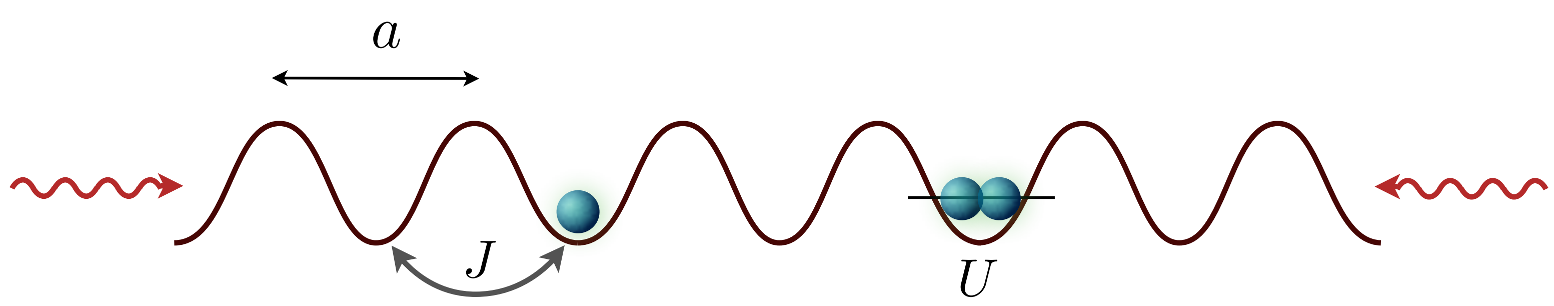
- Indeed, the physics in continuous space is **identical to lattice physics** $\hat{H} = -J\sum_i \left(|i\rangle\langle i+1| + |i+1\rangle\langle i|\right)$

GP equation - Numerical considerations

- **Remark:** Same physics on a real lattice: $\boxed{\textbf{Bose-Hubbard model}}$ e.g. realized in optical lattices

free space (artificial grid)

$a \to 0$ $\Leftrightarrow$



$$\hat{H} = \hat{H}_0 + \frac{g}{2} \int dx\, \hat{\psi}_x^\dagger \hat{\psi}_x^\dagger \hat{\psi}_x \hat{\psi}_x$$

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

- **Hilbert space size:** N particles on M sites:

$$D = \frac{(M+N-1)!}{(M-1)!N!}$$

Examples:

$M = N = 16$	$D \approx 3 \times 10^8$	$\approx 5\,\text{GB}$
$M = 100, N = 5$	$D \approx 1 \times 10^8$	$\approx 1.5\,\text{GB}$
$M = 20, N = 20$	$D \approx 7 \times 10^{10}$	$\approx 10^3\,\text{GB}$

- … in the mean-field approximation: State-vector size is only $\tilde{D} = N$

$$\begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{M-1} \\ \psi_M \end{pmatrix}$$

We can treat huge systems!

- … but we pay a price: **I. We made a strong approximation, II. The equations are now non-linear**

Runge-Kutta Methods: 4th order applied to GP

$$\boxed{\frac{d}{dt}\psi(x,t) = -\mathrm{i}\left(\underbrace{\frac{\hat{p}^2}{2m} + V(x)}_{\hat{H}_1} + g|\psi(x,t)|^2\right)\psi(x,t)}$$

function f … applies the whole RHS of the GP equation

```julia
function gprk4(H1, g, psi, h)

    function f(psi)
        return -1im .* (H1 * psi  +  g .* abs.(psi).^2 .* psi)
    end

    h2 = h/2
    k1 = f(psi)
    k2 = f(psi .+ h2 .* k1)
    k3 = f(psi .+ h2 .* k2)
    k4 = f(psi .+ h .* k3)
    psi += (h/6) .* (k1 .+ 2 .* k2 .+ 2 .* k3 .+ k4)

    return psi
end
```

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

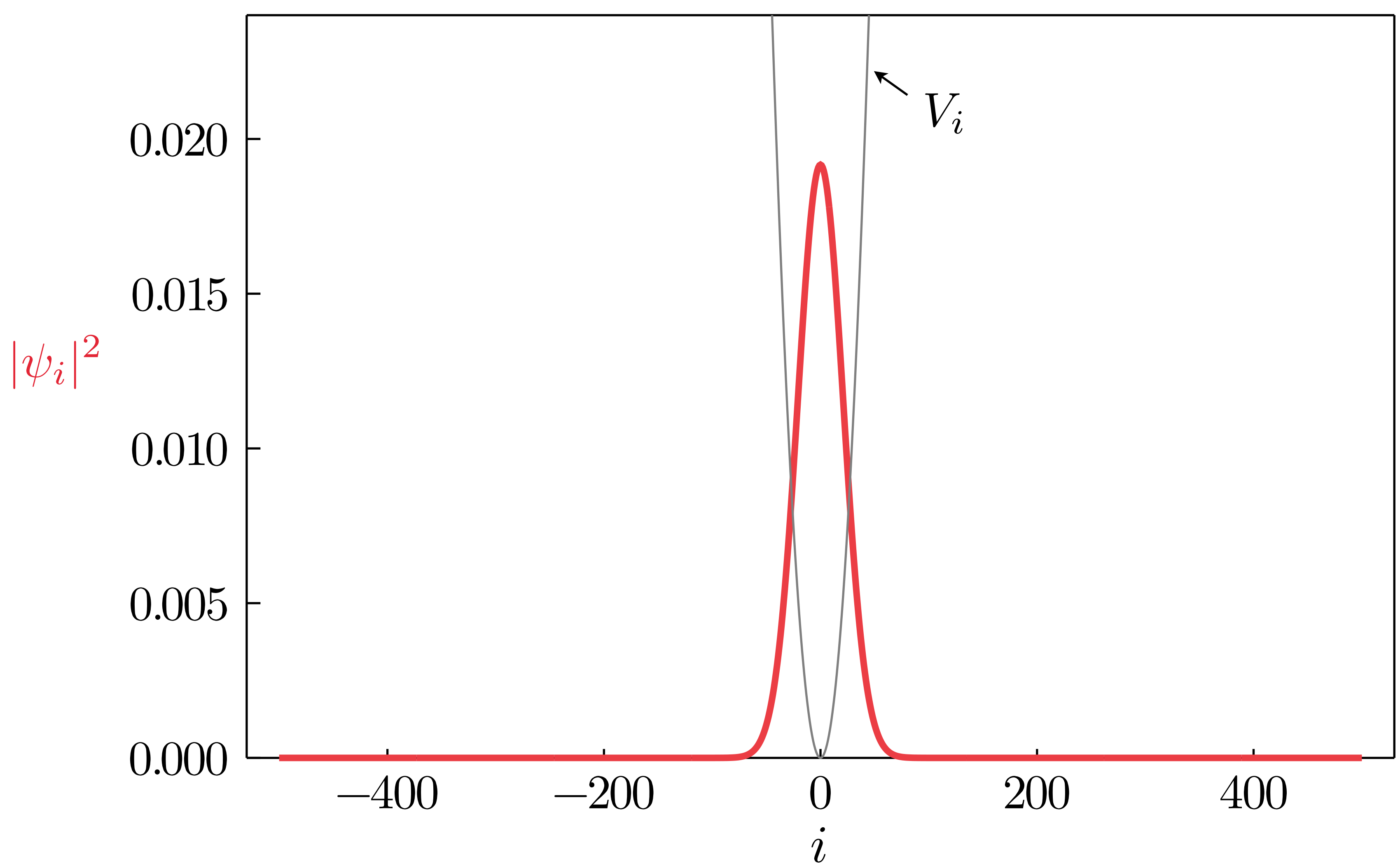
Solving it with RK4

$hJ = 0.02$

1001 grid points

periodic boundaries

- Initial trap: $V_i = 1.2 \times 10^{-5} \times i^2$
- First: Start from ground state ... *we do this by evolving in imaginary time (will see how that works later)*



$g = 0$

No interactions,
we just prepare a standard Gaussian wave-packet
(Standard QM for g=0)

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

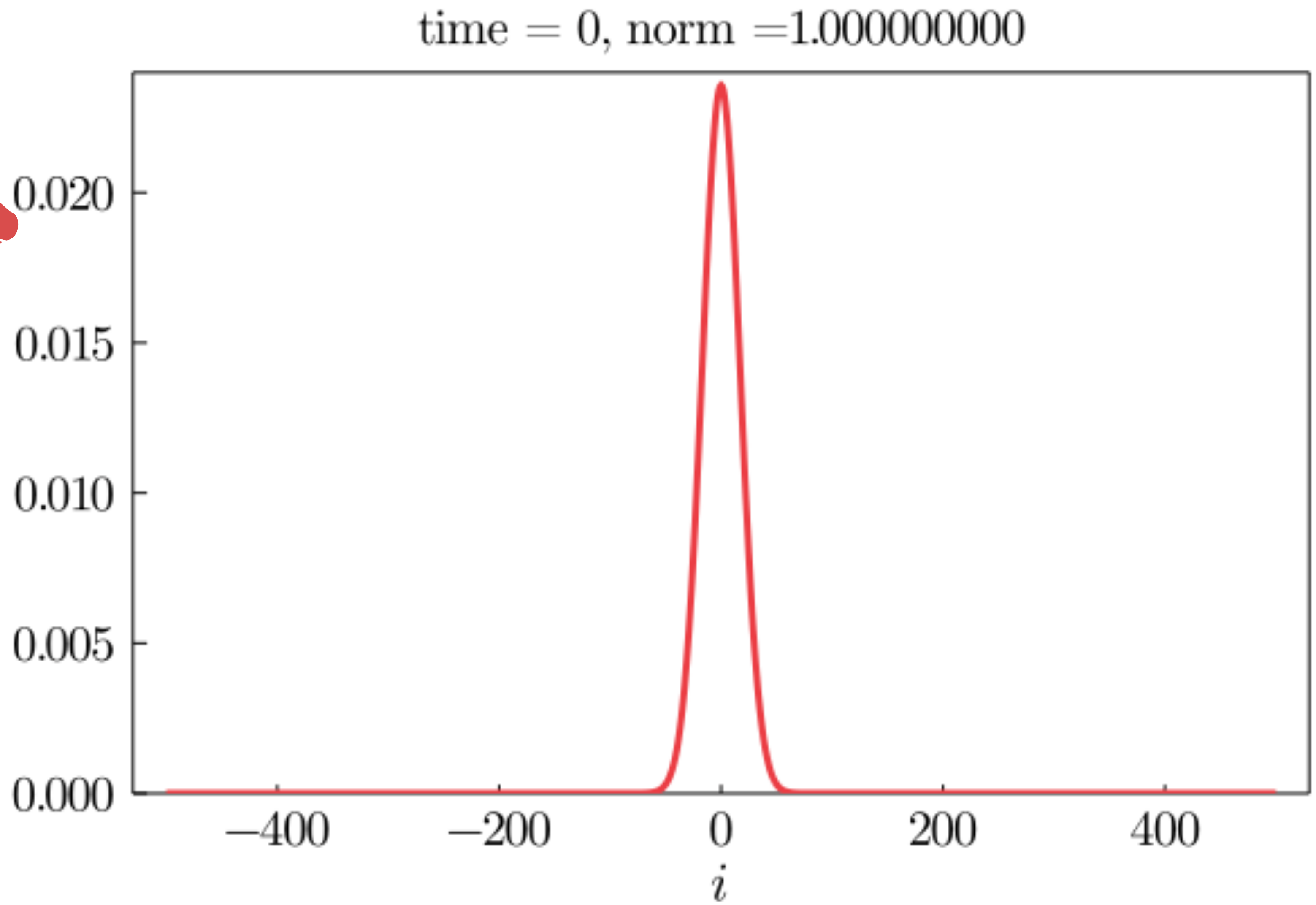
Solving it with RK4

$hJ = 0.02$

1001 grid points

periodic boundaries

- We remove the trap $V_i = 0$
- … and kick the system! This can be done by applying a phase-gradient $\psi_i \to \psi_i e^{\mathrm{i}(ka)i}$



$g = 0$

$ka = 0.2\pi$

Standard evolution of quantum wave-packet with diffusion.

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

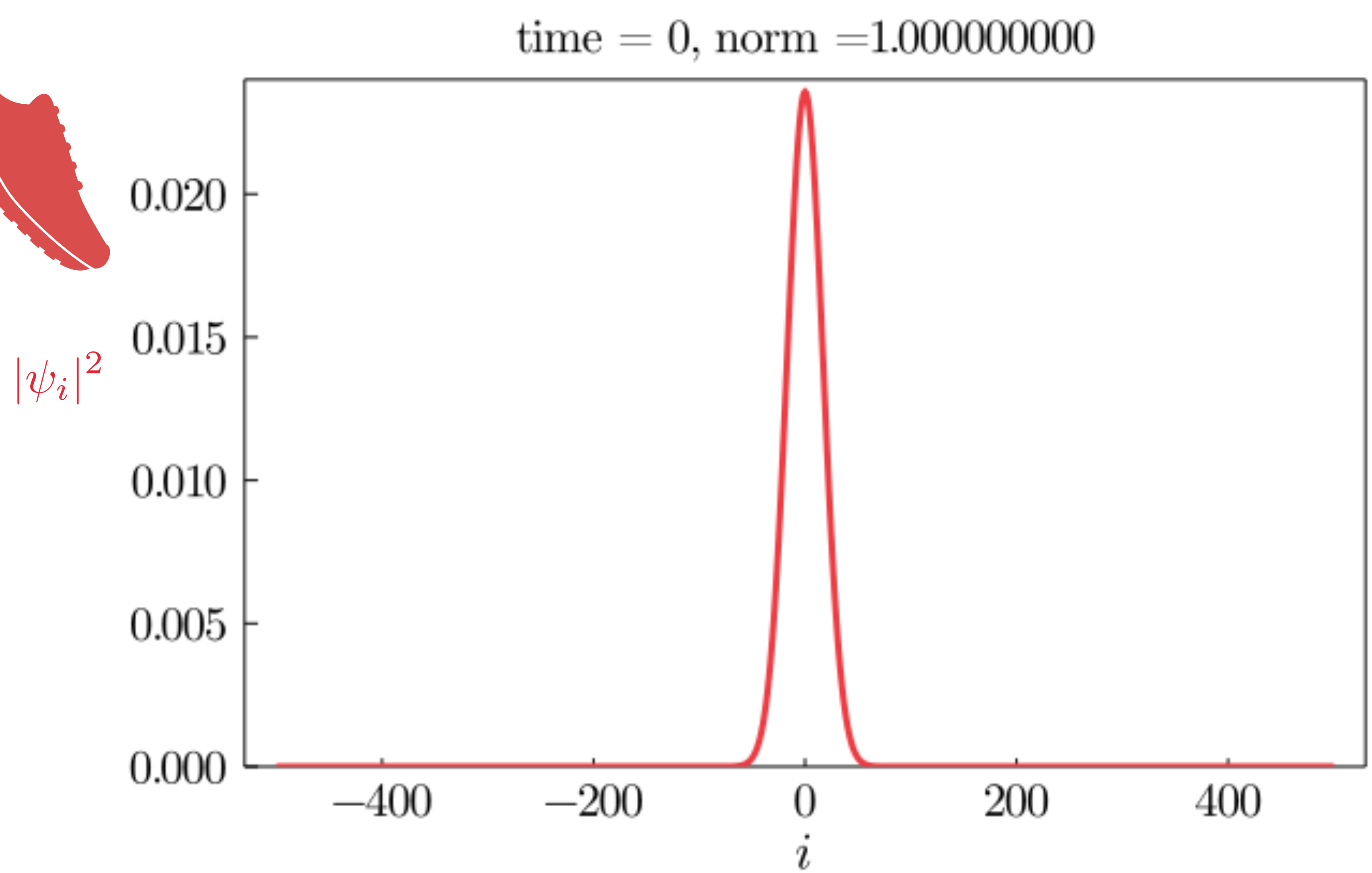
Solving it with RK4

$hJ = 0.02$

1001 grid points

periodic boundaries

- We kick it stronger
- … and kick the system! This can be done by applying a phase-gradient $\psi_i \to \psi_i \mathrm{e}^{\mathrm{i}(ka)i}$



$g = 0$

$ka = 0.4\pi$

Stronger kick! Faster standard evolution of quantum wave-packet with diffusion.

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

Solving it with RK4

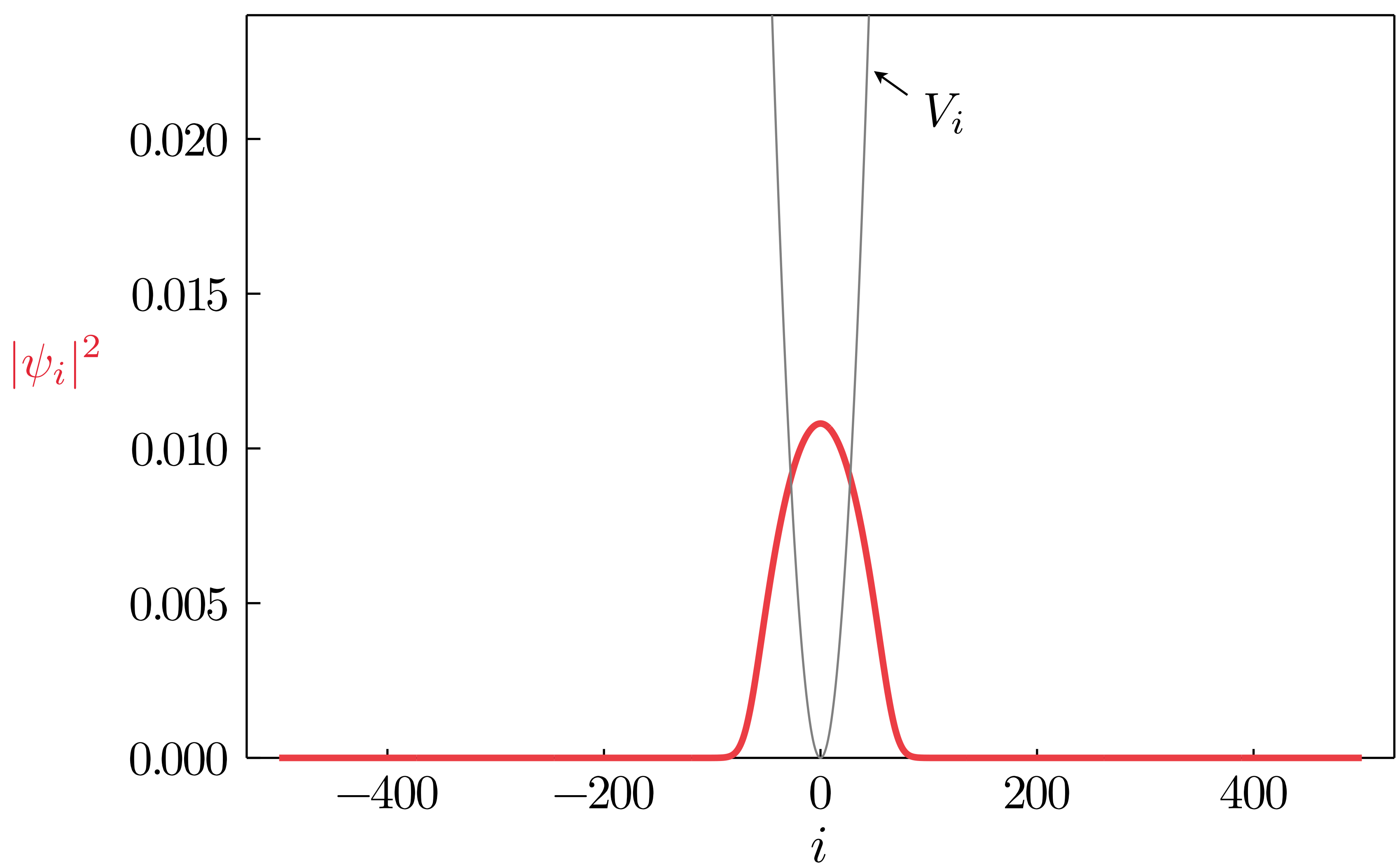
$hJ = 0.02$

1001 grid points

periodic boundaries

- Initial trap: $V_i = 1.2 \times 10^{-5} \times i^2$
- And first we compute the ground state.

…we do this by evolving in imaginary time (will see how that works later)



$g = 5J$

Interactions on!
Much broader classical field for the BEC

Potential takes form of inverted trap -> Result in Thomas-Fermi approximation!

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

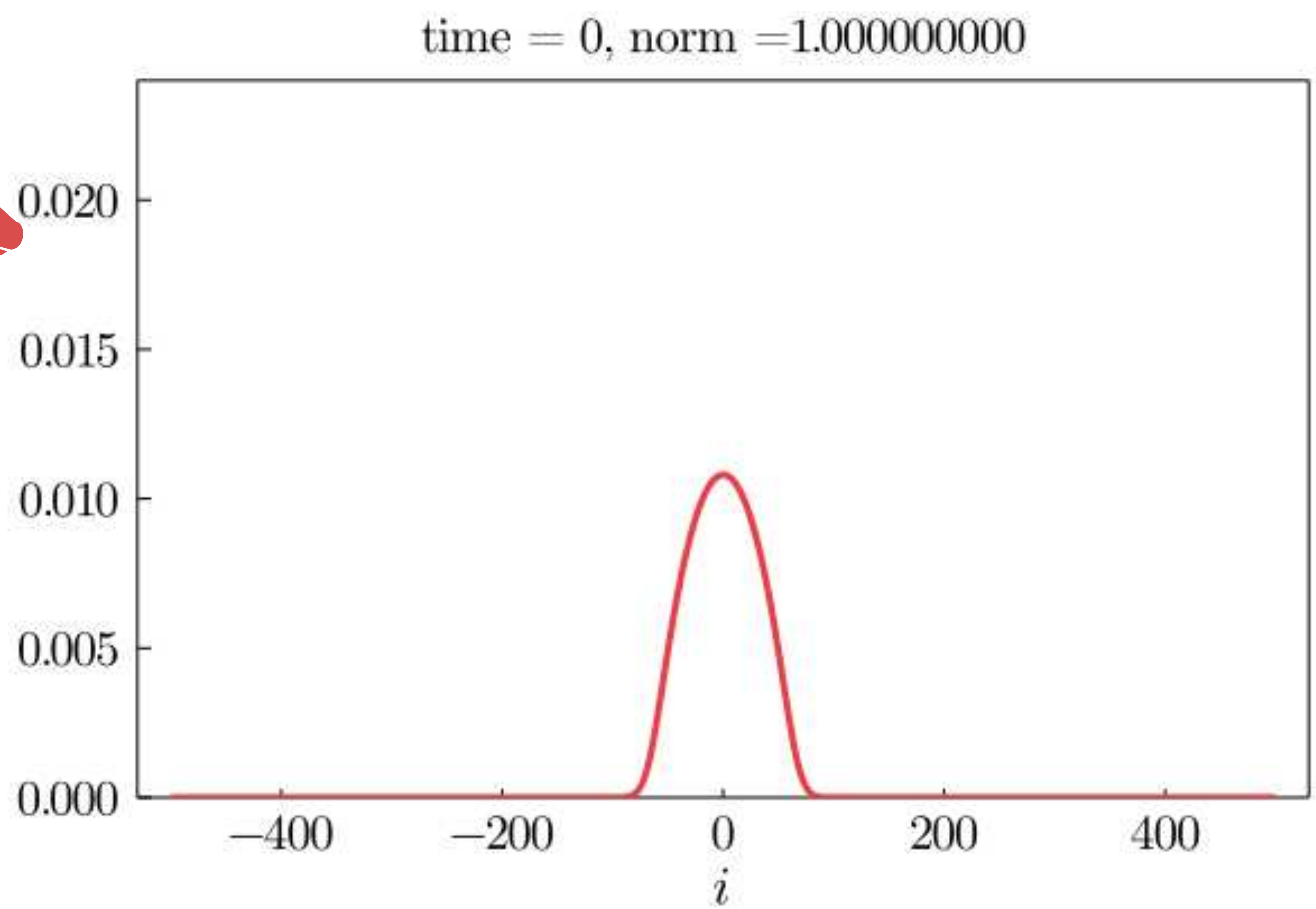
Solving it with RK4

$hJ = 0.02$

1001 grid points

periodic boundaries

- We remove the trap $V_i = 0$
- … and kick the system! This can be done by applying a phase-gradient $\psi_i \to \psi_i e^{\mathrm{i}(ka)i}$



$g = 5J$

$ka = 0.2\pi$

Similar evolution as non-interacting case. Just diffusion. But condensate field keeps the Thomas-Fermi shape (BEC is stable)

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

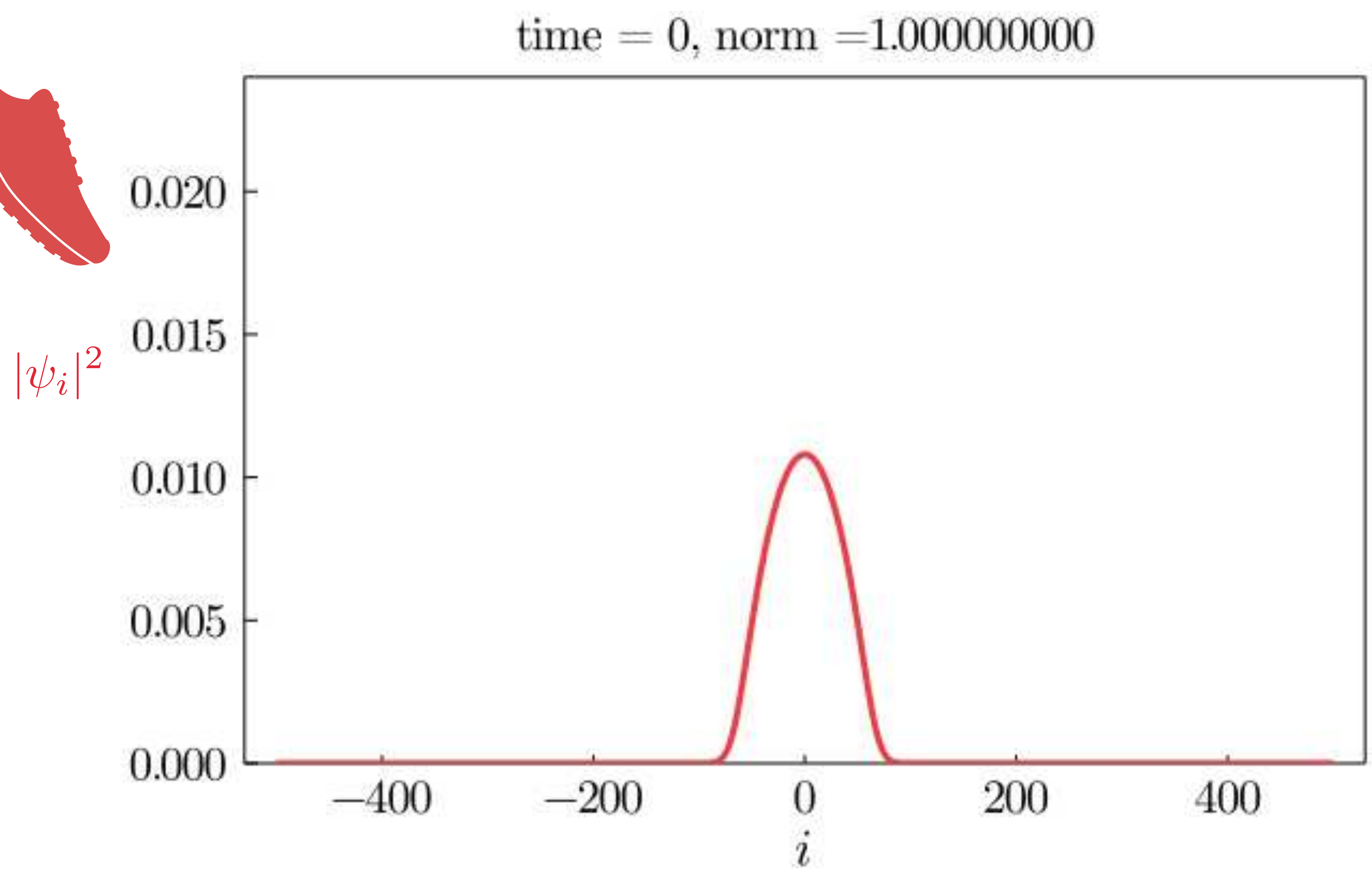
Solving it with RK4

$hJ = 0.02$

1001 grid points

periodic boundaries

- We kick it stronger
- … and kick the system! This can be done by applying a phase-gradient $\psi_i \to \psi_i \mathrm{e}^{\mathrm{i}(ka)i}$



$g = 5J$

$ka = 0.4\pi$

Our BEC get's destroyed!

This is known as dynamical instability!

Runge-Kutta Methods: 4th order applied to GP

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

$$J = \frac{1}{2ma^2} \equiv 1$$

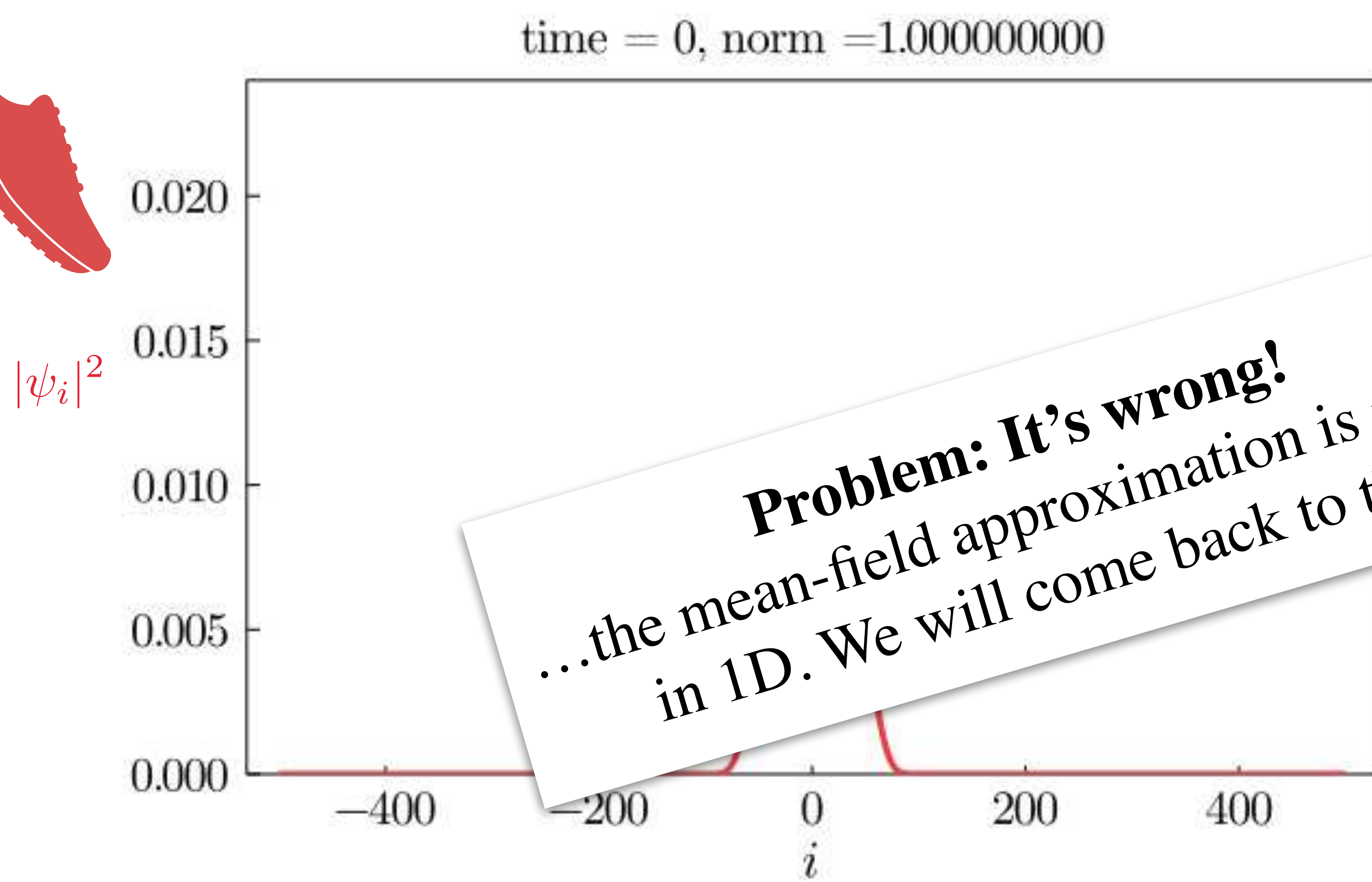
Solving it with RK4

$hJ = 0.02$

1001 grid points

periodic boundaries

- We kick it stronger
- … and kick the system! This can be done by applying a phase-gradient $\psi_i \to \psi_i \mathrm{e}^{\mathrm{i}(ka)i}$



Problem: It's wrong!

…the mean-field approximation is terrible in 1D. We will come back to this.

$g = 5J$

$ka = 0.4\pi$

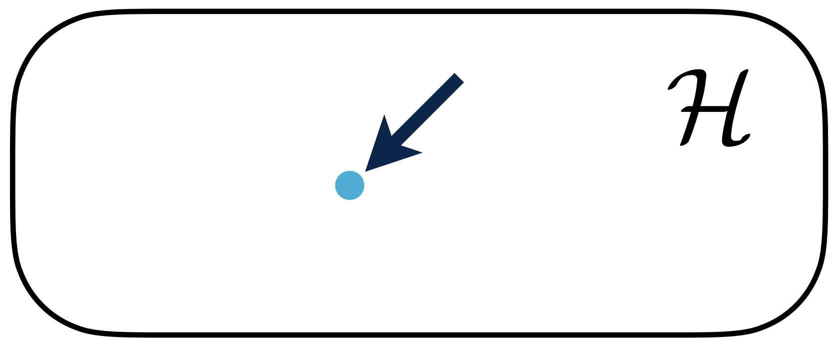
BEC get's destroyed!

This is known as dynamical instability!

See e.g.
New J. Phys. 12, 025014 (2010)

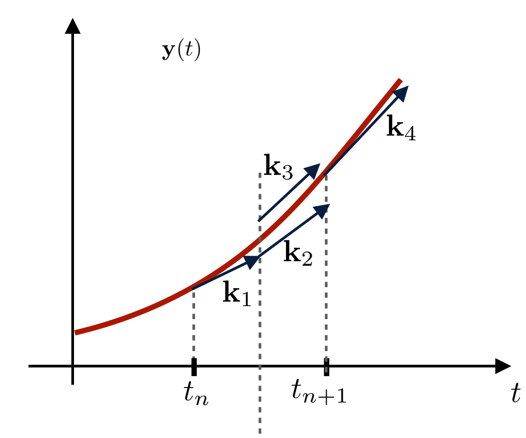
Recap

- Not all quantum problems are linear. Also in quantum mechanics, when making e.g. a mean-field approximation (e.g. a product state approximation for spin models, or in the 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). The model is essentially equivalent to “classical spins”, entanglement is neglected.



- 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).

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



- As example mean-field problem we showed how to simulate dynamics of a transverse Ising model in mean-field and compared to exact results. Generally mean-field works well in low-excitation regimes and for high connectivity