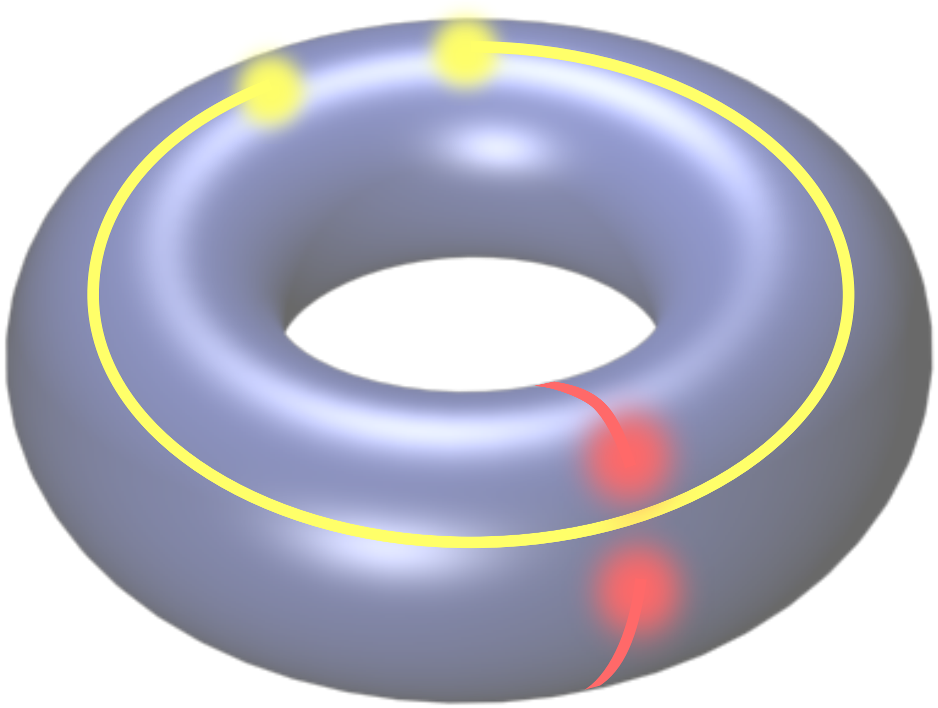
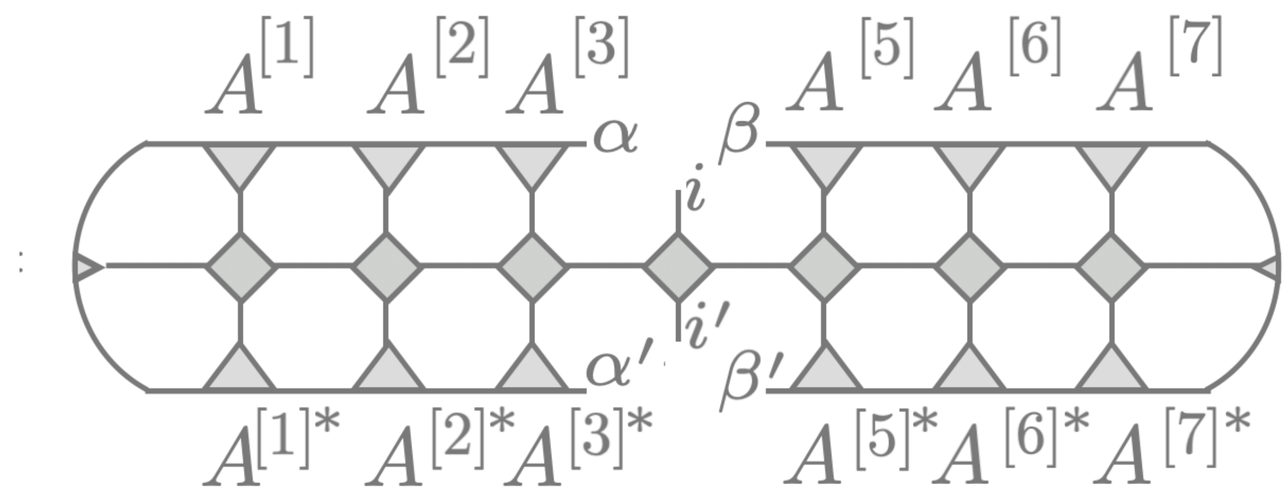
Efficient simulation of interacting topological matter

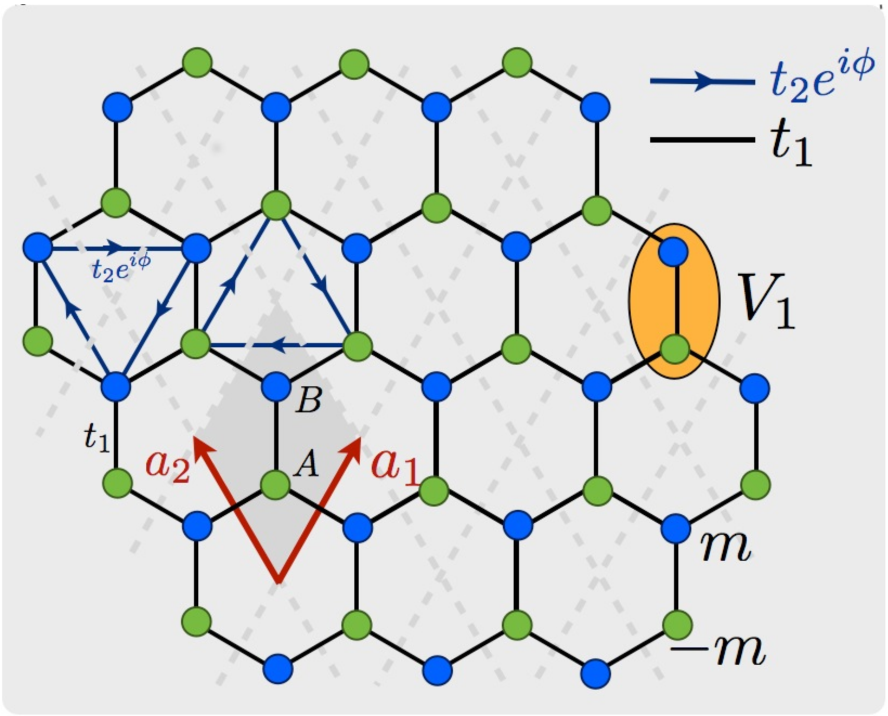
Frank Pollmann (TUM)
Mike Zaletel (UC Berkeley)

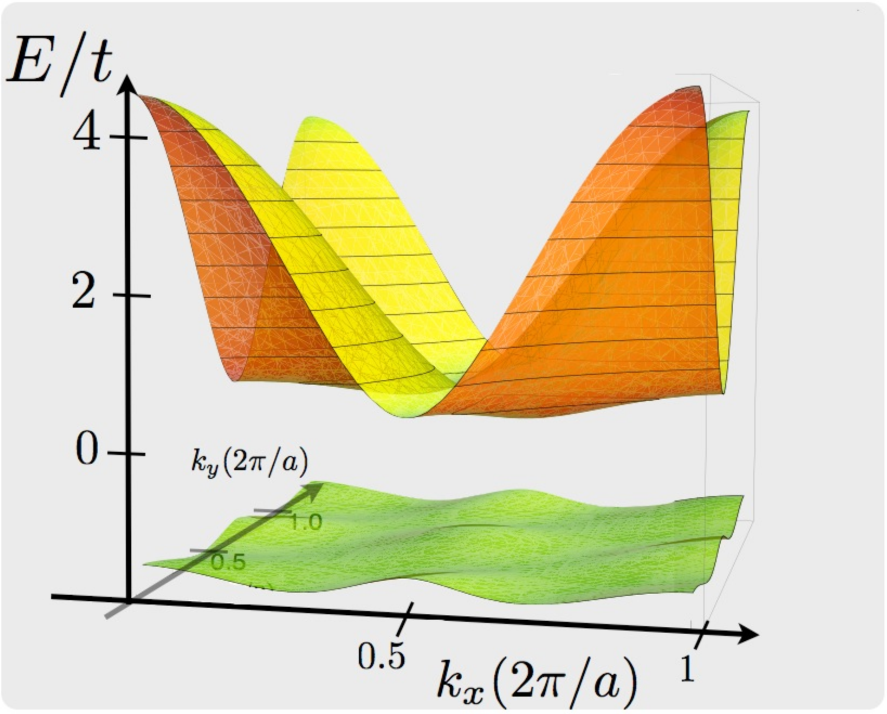




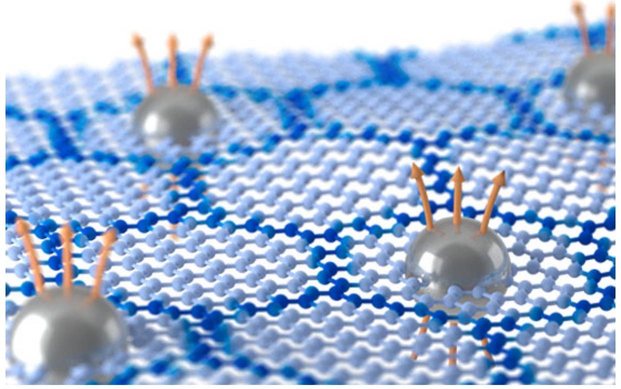
Topological Matter School 2019
San Sebastian

Efficient simulation of interacting topological matter

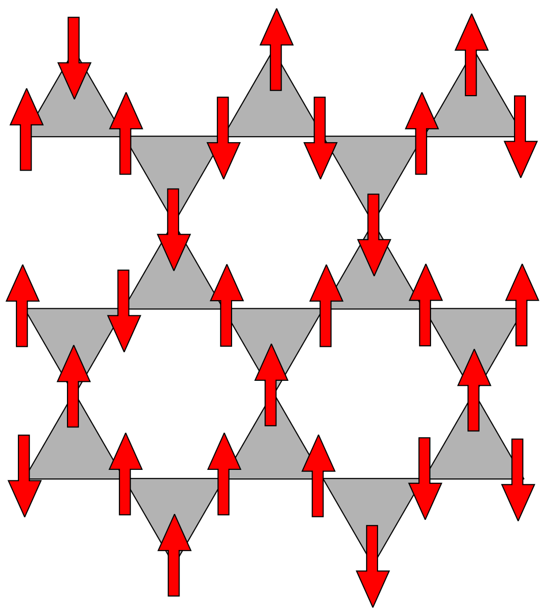




What is the nature of the
many body ground state $|\psi_0\rangle$?



Efficient simulation of interacting topological matter



$$H = \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j$$

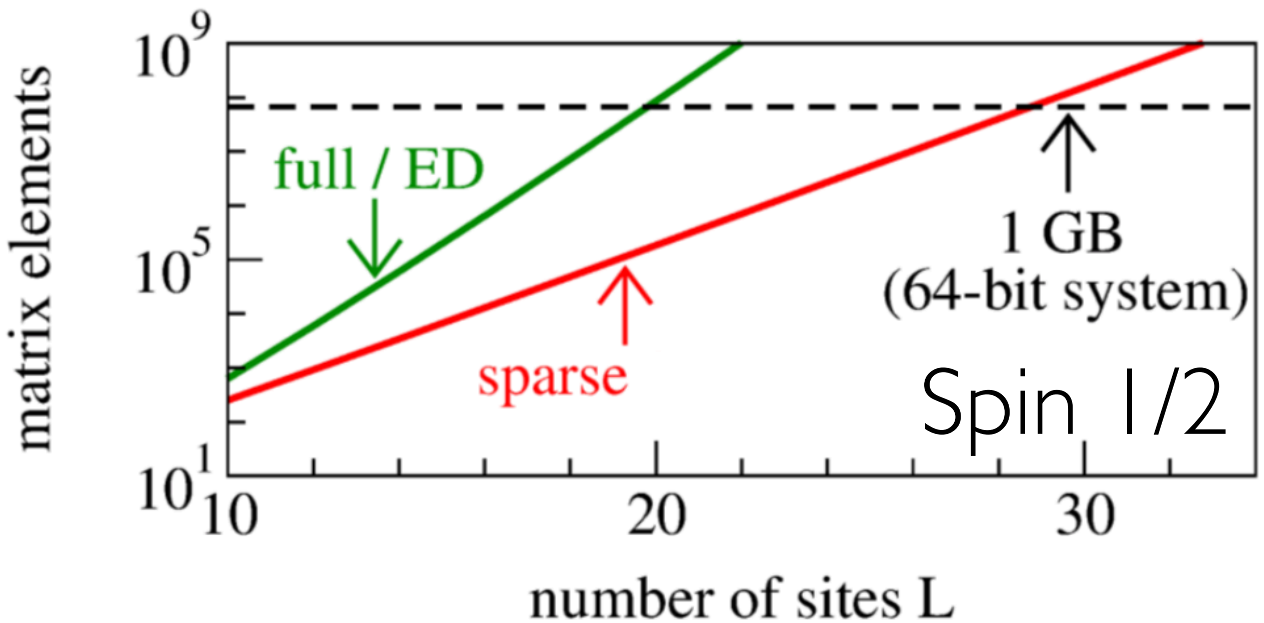
Herbertsmithite

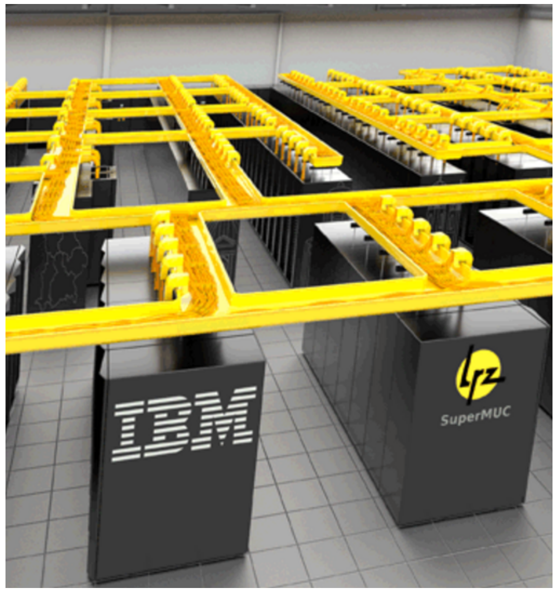
What is the nature of the many body ground state $|\psi_0\rangle$?

Complexity of a quantum many-body problem

Many-body Hilbert space

$$|\psi\rangle = \sum_{j_1,j_2,\ldots,j_L} \psi_{j_1,j_2,\ldots,j_L} |j_1\rangle|j_2\rangle \ldots |j_L\rangle \ , \ j_n = 1\ldots d$$





➡ Full diagonalization up to ~20 sites

➡ Sparse methods up to ~30 sites

Matrix-Product States

Many-body Hilbert space

$$|\psi\rangle = \sum_{j_1,j_2,\ldots,j_L} \psi_{j_1,j_2,\ldots,j_L} |j_1\rangle|j_2\rangle\ldots|j_L\rangle \ , \ j_n = 1\ldots d$$

Matrix-Product States: Reduction of #variables $d^L \to Ld\chi^2$

$$\psi_{j_1,j_2,\ldots,j_L} \approx \sum_{\alpha_1,\alpha_2,\ldots,\alpha_{L-1}} A^{j_1}_{\alpha_1} A^{j_2}_{\alpha_1,\alpha_2} \ldots A^{j_L}_{\alpha_{L-1}} \qquad \alpha_j = 1\ldots\chi$$

Once we have an MPS representation, we can calculate (almost) everything exactly!

Why is it a good representation?

Efficient simulation of interacting topological matter

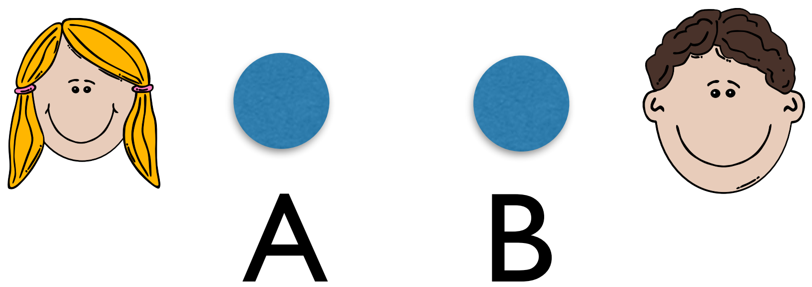
I) Entanglement and Matrix-Product States

II) Time Evolving Block Decimation

III) Density-Matrix Renormalization Group

IV) Extracting topological invariants

Entanglement



Product state (=non-entangled):

$$|\psi\rangle = \frac{1}{2}\Big(|\uparrow\rangle_A + |\downarrow\rangle_A\Big)\Big(|\uparrow\rangle_B + |\downarrow\rangle_B\Big)$$

Entangled state

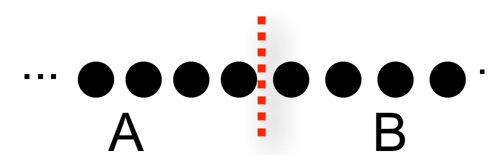
$$|\psi\rangle = \frac{1}{\sqrt{2}}\Big(|\uparrow\rangle_A|\downarrow\rangle_B + |\downarrow\rangle_A|\uparrow\rangle_B\Big)$$

Entanglement

A generic quantum state has a d^L dimensional Hilbert space

$$|\psi\rangle = \sum_{j_1, j_2, \ldots, j_L} \psi_{j_1, j_2, \ldots, j_L} |j_1\rangle |j_2\rangle \ldots |j_L\rangle \ , \ j_n = 1 \ldots d$$

Decompose a state into a superposition of product states (**Schmidt decomposition**)



$$|\psi\rangle = \sum_{i,j} C_{i,j} |i\rangle_A \otimes |j\rangle_B = \sum_{\alpha} \Lambda_\alpha |\alpha\rangle_A \otimes |\alpha\rangle_B, \ \langle\alpha|\alpha'\rangle = \delta_{\alpha\alpha'}$$

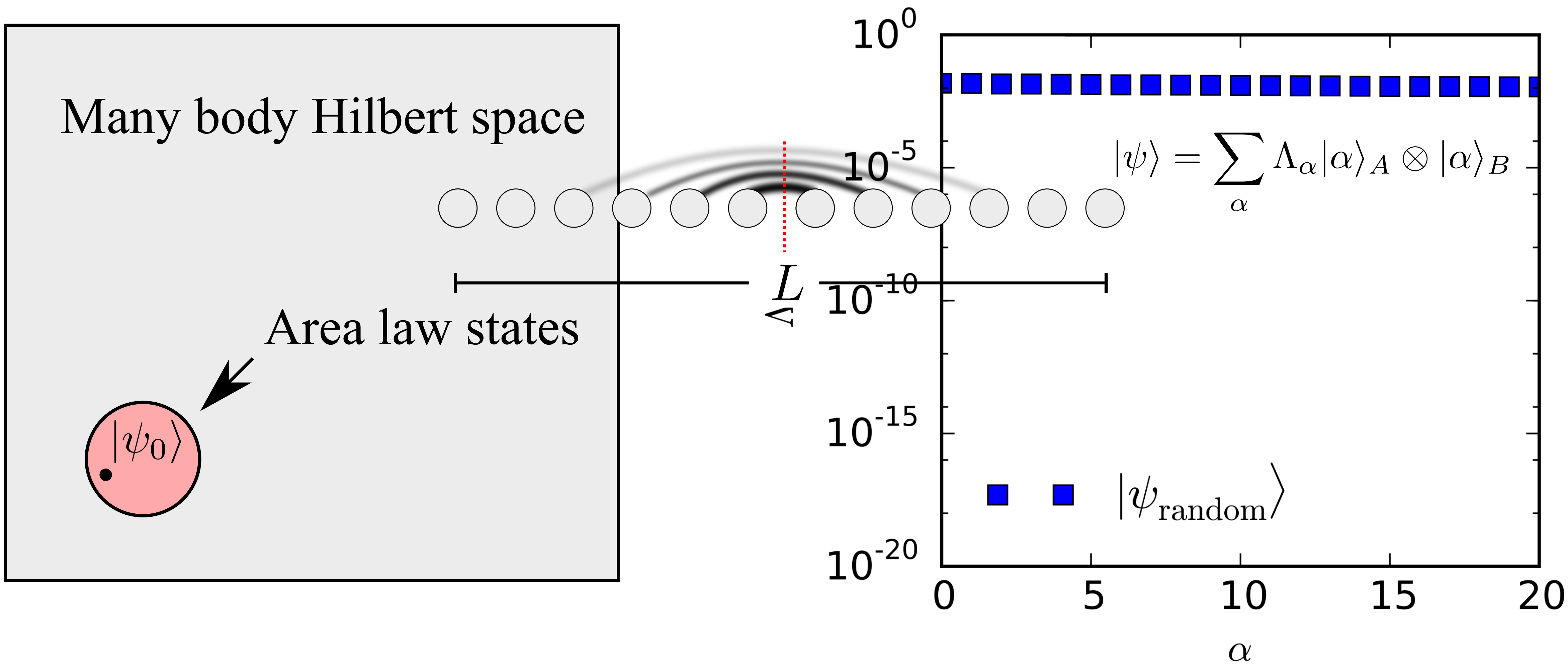
Entanglement entropy as a measure for the amount of entanglement $S = -\sum_\alpha \Lambda_\alpha^2 \log \Lambda_\alpha^2$

(Equivalent to $S = -\mathrm{Tr} \rho_A \log \rho_A$ with $\rho_A = \mathrm{Tr}_B |\psi\rangle\langle\psi|$)

Entanglement

Area law for ground states of local (gapped) Hamiltonians in one dimensional systems

$S(L) = \text{const.}$ [Srednicki '93, Hastings '07]



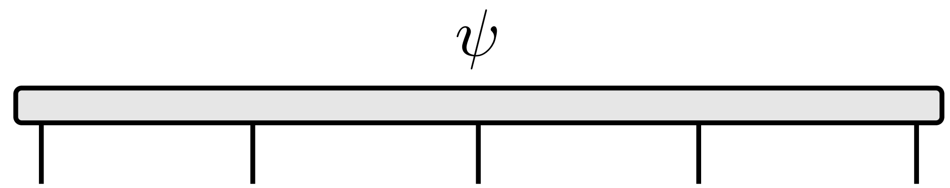
All ground states live in a tiny corner of the Hilbert space!

Matrix-Product States

Coefficients in the many-body wave function:

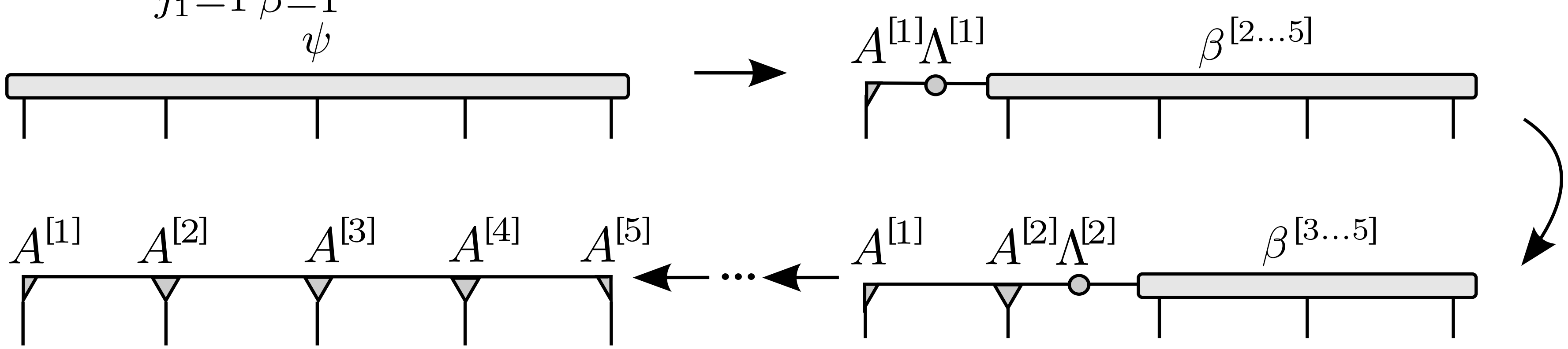
Rank-L tensor: diagrammatic representation

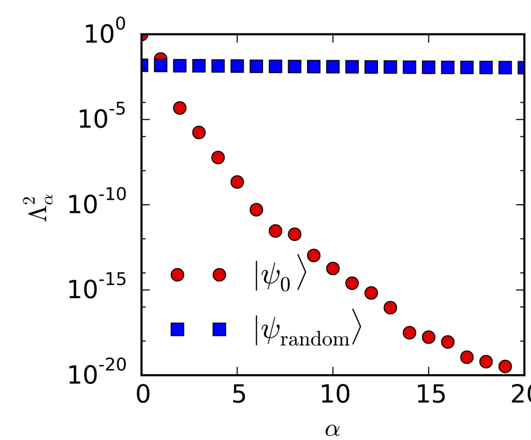
$$\psi_{j_1,j_2,j_3,j_4,j_5} =$$



Successive Schmidt decompositions

$$|\psi\rangle = \sum_{j_1=1}^{d} \sum_{\beta=1}^{d} A_{\beta}^{[1]j_1} \Lambda_{\beta}^{[1]} |j_1\rangle |\beta\rangle_{[2,\ldots N]}$$





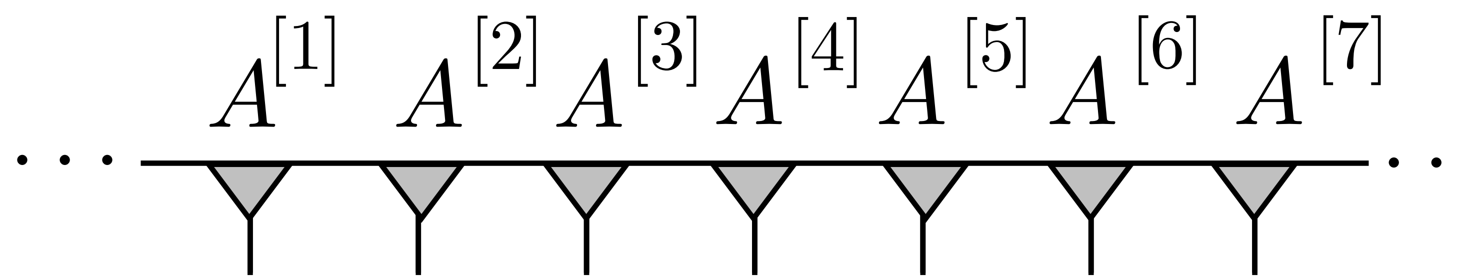
$$\psi_{j_1,j_2,\ldots,j_L} \approx \sum_{\alpha_1,\alpha_2,\ldots,\alpha_{L-1}} A_{\alpha_1}^{j_1} A_{\alpha_1,\alpha_2}^{j_2} \ldots A_{\alpha_{L-1}}^{j_L}$$

MPS are tailored to describe 1D systems with an area law!

MPS and the canonical form

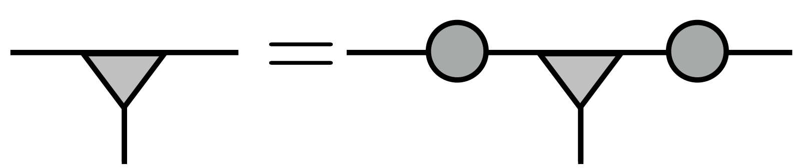
From now on: Leave out site indices!

$|\psi\rangle :$



MPS is not unique

$$\tilde{A}^{i_n} \quad = \quad X A^{i_n} X^{-1}$$



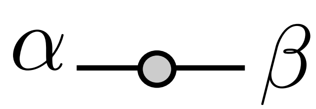
➡ $\tilde{A}^{i_n}$ describes the same state!

MPS and the canonical form

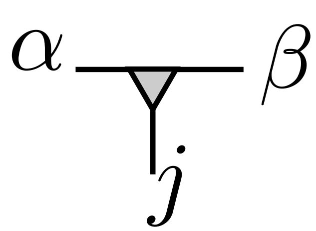
Choose a convenient representation in **Canonical Form**:
Bond index corresponds to Schmidt decomposition! [Vidal '03]

$$|\psi\rangle = \sum_{\alpha=1}^{\chi} \Lambda_\alpha |\alpha\rangle_L \otimes |\alpha\rangle_R \qquad \text{with} \qquad \langle\alpha|\alpha'\rangle = \delta_{\alpha\alpha'}$$

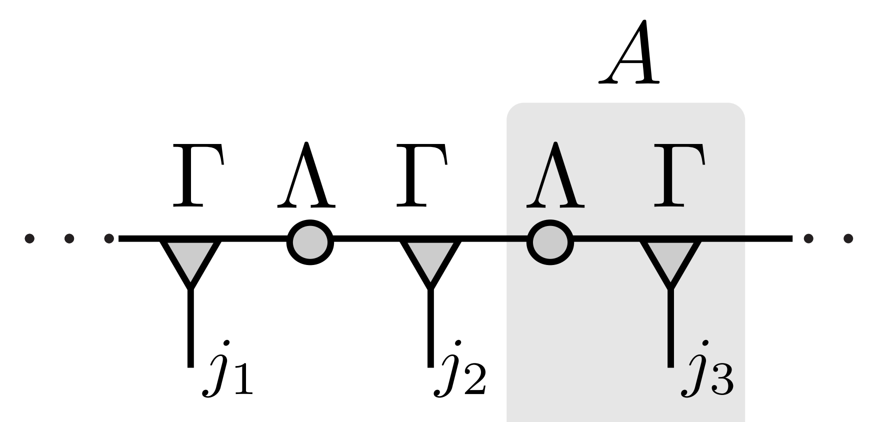
Write tensor $A^{i_n}_{\alpha\beta}$ as product of

$\Lambda_{\alpha\beta} =$ 

: Diagonal matrix with Schmidt values

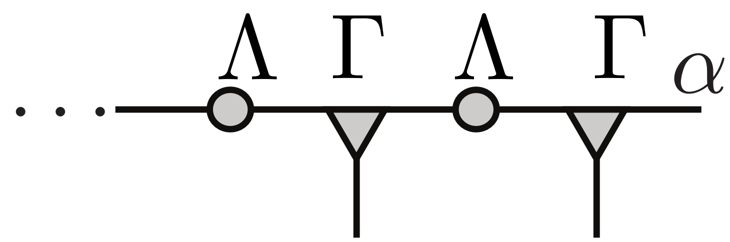
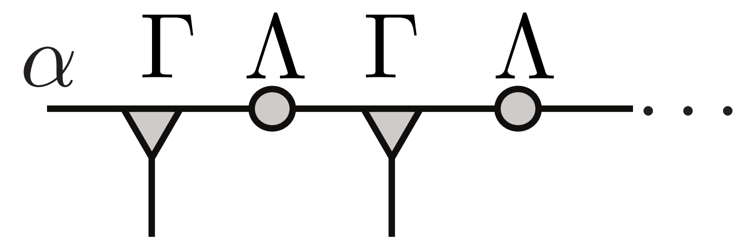
$\Gamma^{j}_{\alpha\beta} =$ 

: Tensor relating to Schmidt basis

$\psi_{\ldots,\, j_1, j_2, j_3, \ldots} =$ 

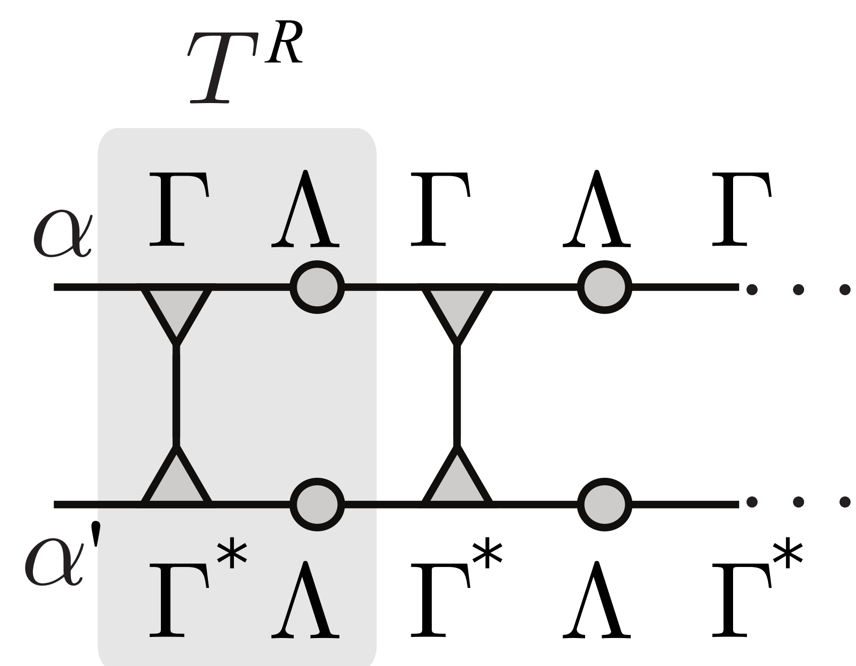
MPS and the canonical form

Schmidt states in terms of the MPS:

$|\alpha\rangle_L =$  $\quad |\alpha\rangle_R =$ 

Orthogonality:

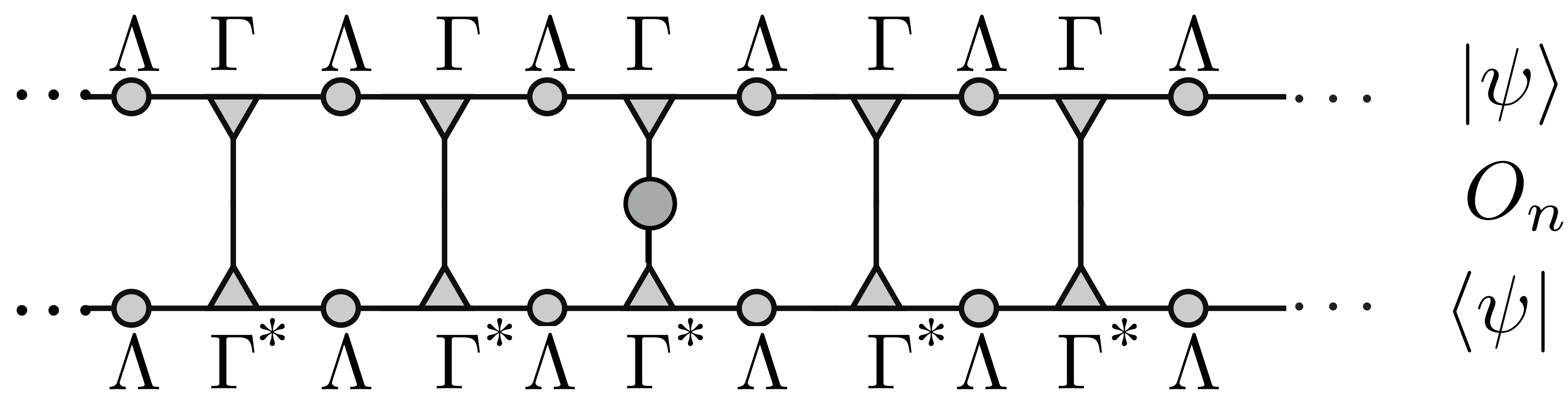
$$\langle \alpha' | \alpha \rangle_R = \delta_{\alpha'\alpha} \equiv$$



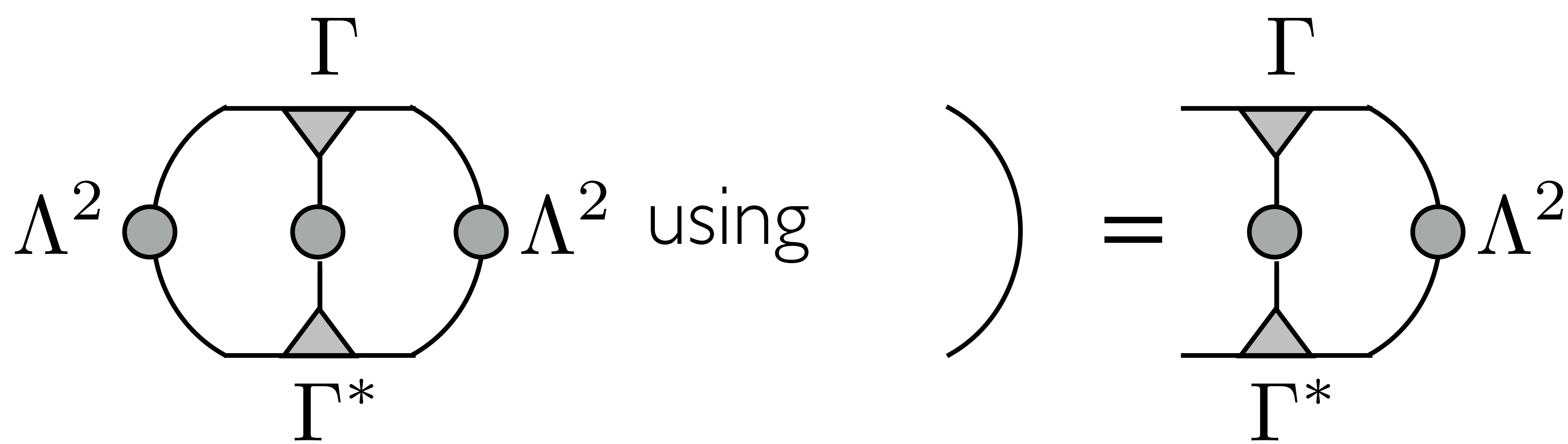
MPS and the canonical form

Efficient evaluation of **expectation values**:

$\langle \psi | O_n | \psi \rangle =$



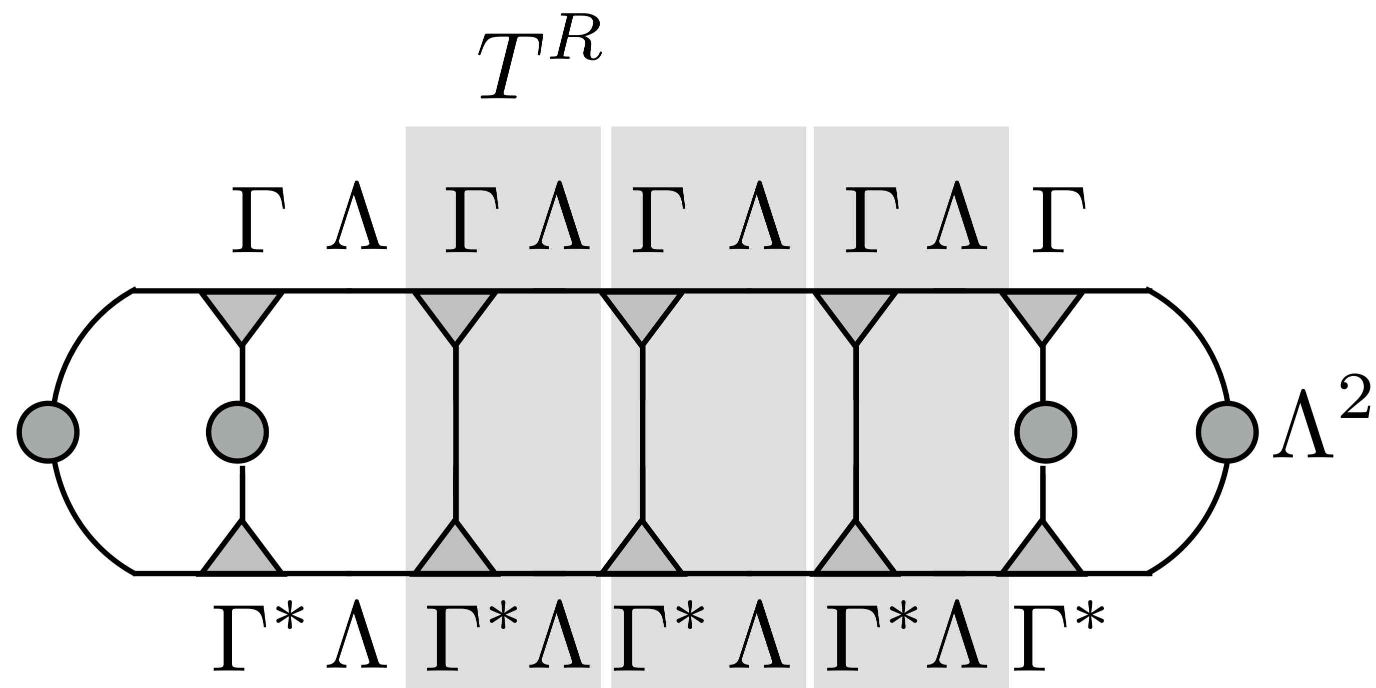
$=$



MPS and the canonical form

Efficient evaluation of **correlation functions**:

$$\langle\psi|O_mO_n|\psi\rangle = \Lambda^2$$

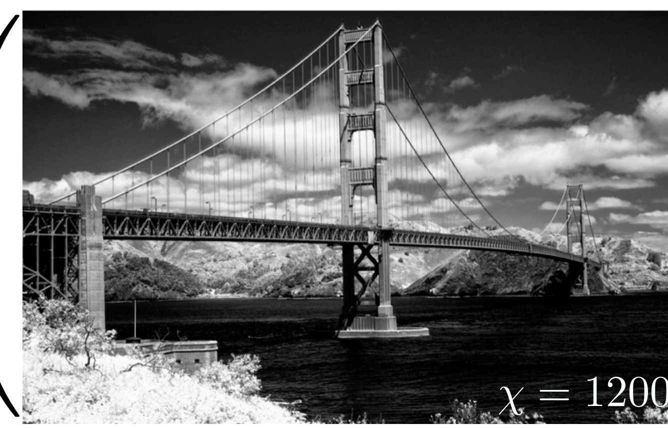


Tutorial: (A) SVD Compression

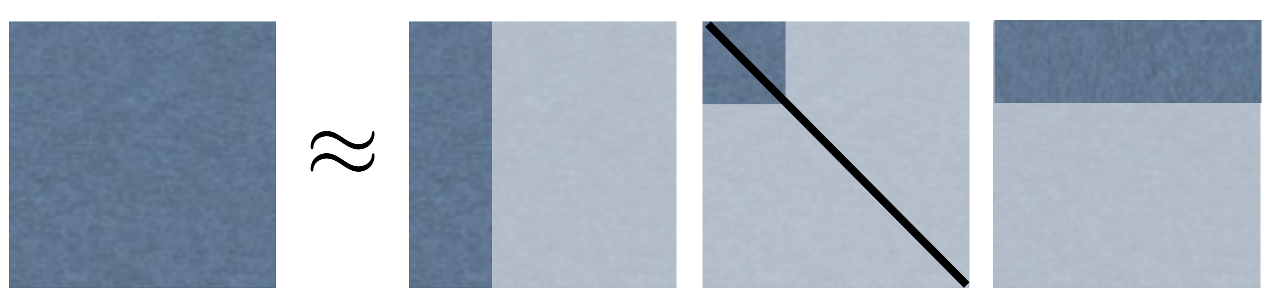
Example: $$|\psi\rangle = \sum_{i=1}^{N_A}\sum_{j=1}^{N_B} C_{ij}|i\rangle_A|j\rangle_B = \sum_{\gamma}\lambda_\gamma|\phi_\gamma\rangle_A|\phi_\gamma\rangle_B$$

Matrix can represent an image (array of pixel)

$$C = \begin{pmatrix} 0.23 & \cdots & 0.56 \\ \vdots & \ddots & \vdots \\ 0.22 & \cdots & 0.34 \end{pmatrix} =$$



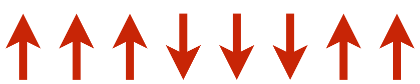
Reconstruction of the matrix (image) from a small number of Schmidt states (SVD):

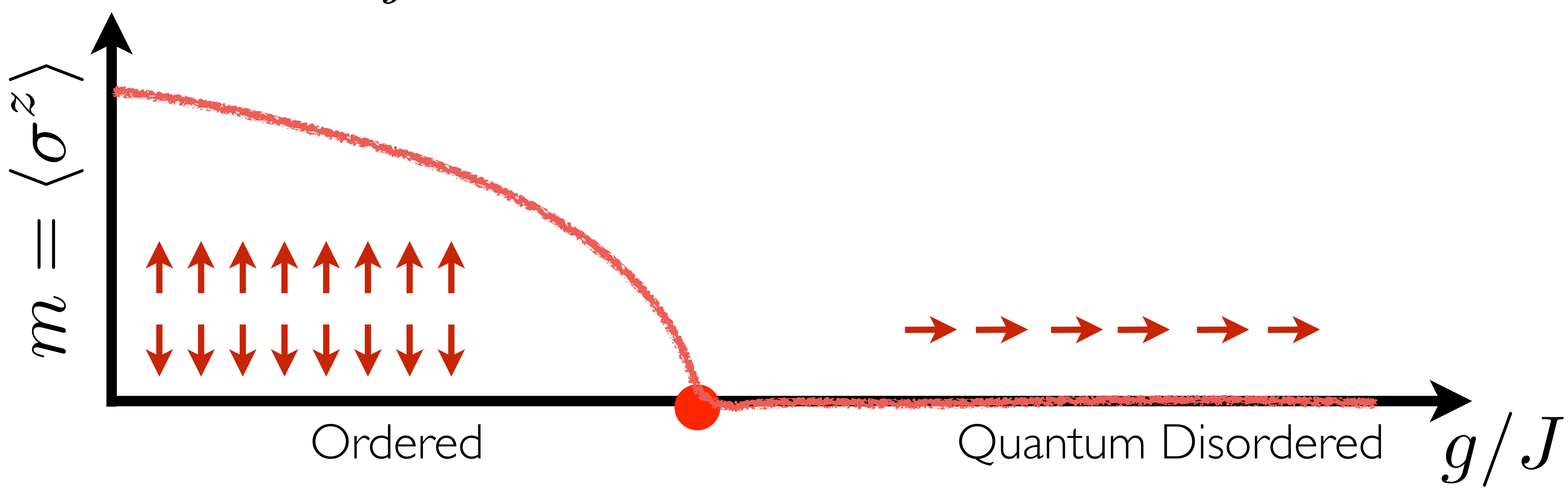


Tutorial: (A) SVD Compression

Quantum phases at $T = 0$: Transverse field Ising model with $\mathbb{Z}_2$ symmetry [Elliott et al.'70]

$$H = -\sum_j (J\sigma_j^z \sigma_{j+1}^z + g\sigma_j^x)$$





Entanglement of the ground state depends on g/J.

Tutorial: (A) SVD Compression

https://tms-dipc.org/hands-on-information/

I) Analyse the Schmidt spectra of the images and how they can be compressed

II) How does the number of singular values needed to represent the image on its size?

III) How does a product state look like as an image?

IV) Try out the different quantum states and study different cuts.

Efficient simulation of interacting topological matter

I) Entanglement and Matrix-Product States

II) Time Evolving Block Decimation

III) Density-Matrix Renormalization Group

IV) Extracting topological invariants

Time evolving block decimation

Assume we have a Hamiltonian of the form

$$H = \sum_j h^{[j,j+1]}$$

Time evolution in real time

$$|\psi_t\rangle = \exp(-iHt)|\psi_{t=0}\rangle$$

Time evolution in imaginary time

$$|\psi_0\rangle = \lim_{\tau\to\infty} \frac{\exp(-H\tau)|\psi_i\rangle}{||\exp(-H\tau)|\psi_i\rangle||}$$

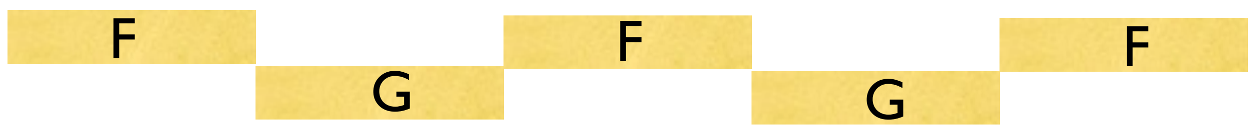
Time evolving block decimation

Consider a Hamiltonian $H = \sum_j h^{[j,j+1]}$ [Vidal '03]

Decompose the Hamiltonian as $\mathrm{H{=}F{+}G}$

$$F \equiv \sum_{\text{even } j} F^{[j]} \equiv \sum_{\text{even } j} h^{[j,j+1]}$$

$$G \equiv \sum_{\text{odd } j} G^{[j]} \equiv \sum_{\text{odd } j} h^{[j,j+1]}$$



We observe $[F^{[r]}, F^{[r']}] = 0$ $([G^{[r]}, G^{[r']}] = 0)$
but $[G, F] \neq 0$

Time evolving block decimation

Apply Suzuki-Trotter decomposition of order p

$$\exp\left(-i(F+G)\delta t\right) \approx f_p\left[\exp(-F\delta t), \exp(-G\delta t)\right]$$

with $f_1(x,y) = xy$, $f_2(x,y) = x^{1/2} y x^{1/2}$, etc.

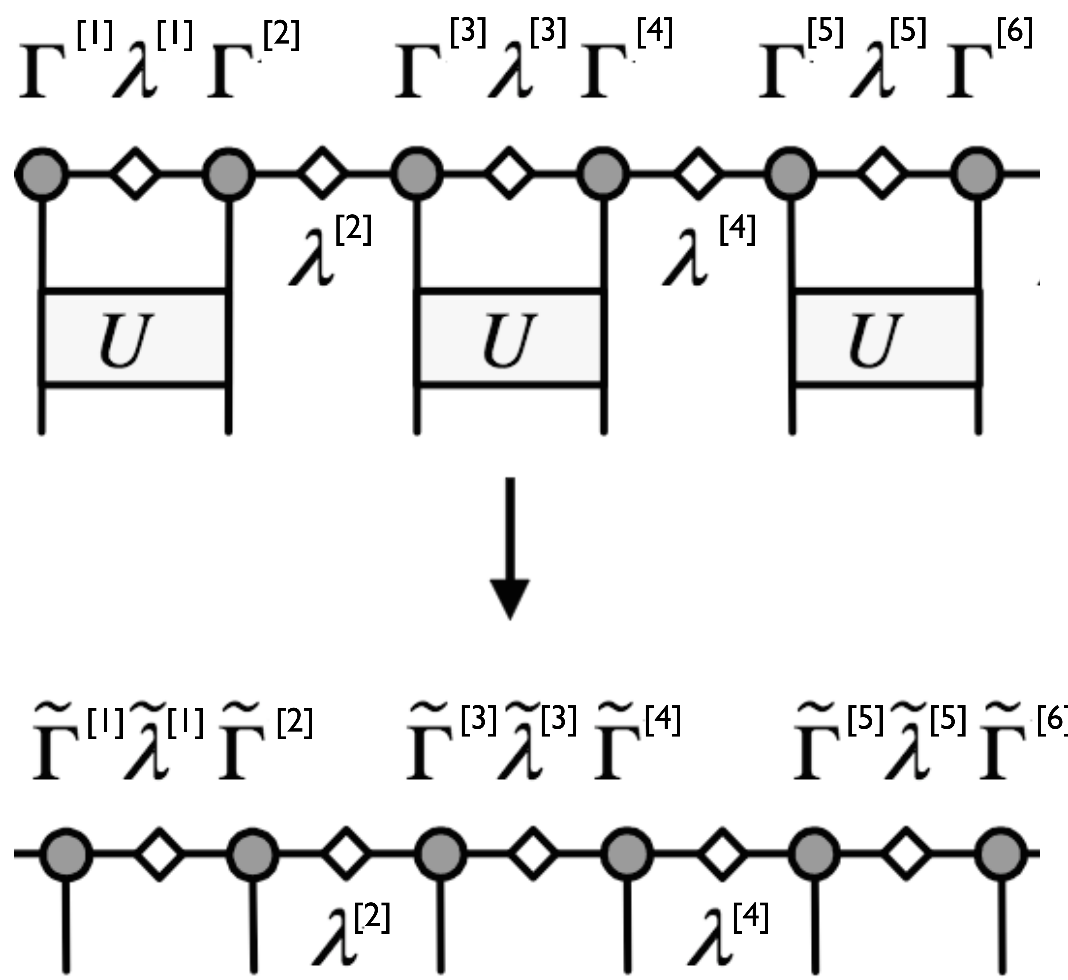
Two chains of two-site gates

$$\begin{aligned} U_F &= \prod_{\text{even } r} \exp(-iF^{[r]}\delta t) \\ U_G &= \prod_{\text{odd } r} \exp(-iG^{[r]}\delta t) \end{aligned}$$

Each gate affects the state only locally

Time evolving block decimation

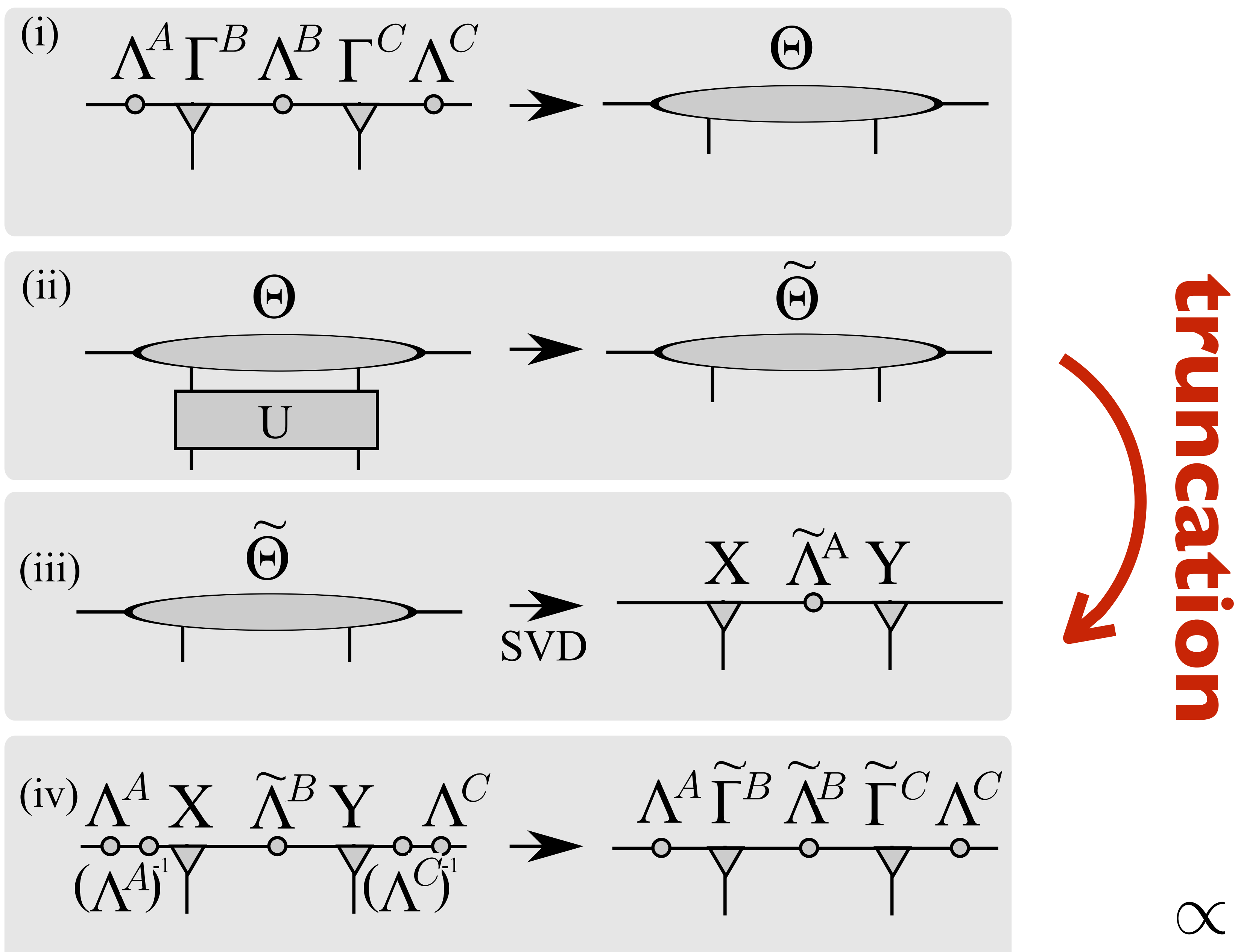
Time Evolving Block Decimation algorithm (TEBD)



How do we get the original form back?

Time evolving block decimation

Time Evolving Block Decimation (TEBD) algorithm [Vidal '03]



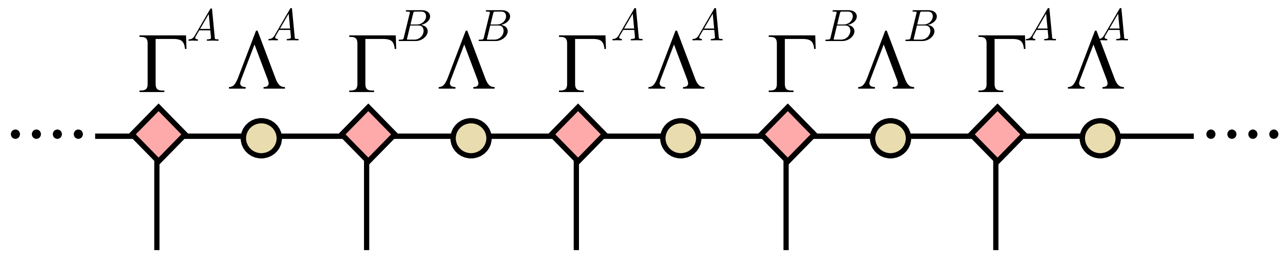
$\propto d^3 \chi^3$

Translationally invariant, infinite systems!

Assume that $|\psi\rangle$ is translational invariant and $L = \infty$:
infinite TEBD (iTEBD)

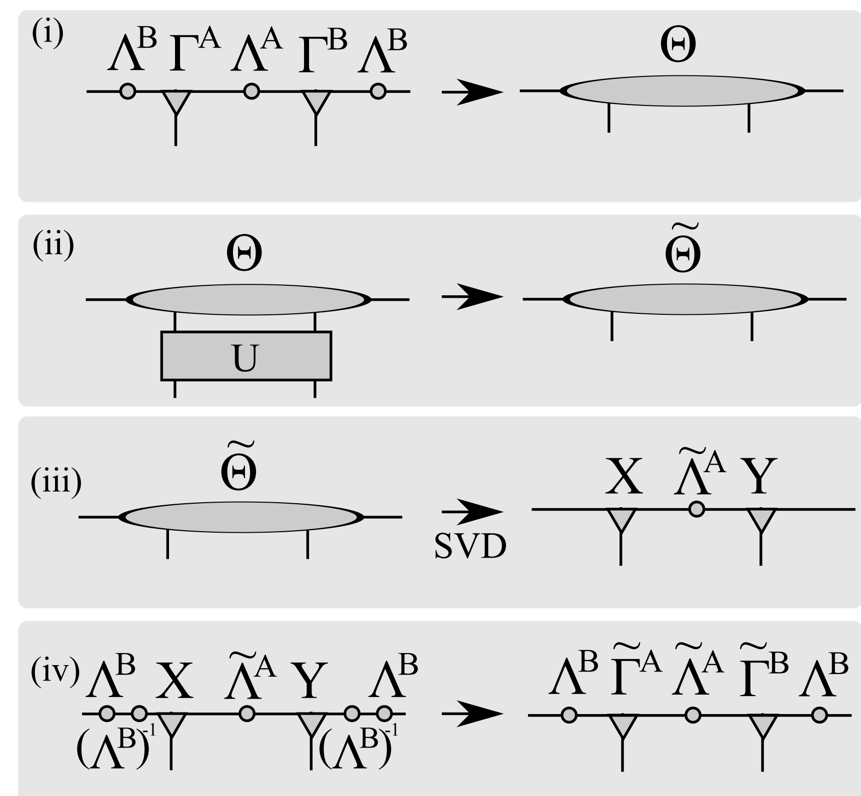
Partially break translational symmetry to simulate the action of the gates

$$\Gamma^{[2r]} = \Gamma^{A},\ \lambda^{[2r]} = \lambda^{A},\ \Gamma^{[2r+1]} = \Gamma^{B},\ \lambda^{[2r+1]} = \lambda^{B}$$



iTEBD algorithm

```
import numpy as np

# First define the parameters of the model / simulation
J=1; g=0.5; chi=10; d=2; delta=0.01; N=1000; G = np.random.rand(2,d,chi,chi);l = np.random.rand(2,chi)

# Generate the two-site time evolution operator
H = np.array( [[J,-g/2,-g/2,0], [-g/2,-J,0,-g/2], [-g/2,0,-J,-g/2], [0,-g/2,-g/2,J]] )
w,v = np.linalg.eig(H)
U = np.reshape(np.dot(np.dot(v,np.diag(np.exp(-delta*(w)))),np.transpose(v)),(2,2,2,2))

# Perform the imaginary time evolution alternating on A and B bonds
for step in range(0, N):
    A = np.mod(step,2); B = np.mod(step+1,2)

    # Construct theta
    theta = np.tensordot(np.diag(l[B,:]),G[A,:,:,:],axes=(1,1))
    theta = np.tensordot(theta,np.diag(l[A,:],0),axes=(2,0))
    theta = np.tensordot(theta,G[B,:,:,:],axes=(2,1))
    theta = np.tensordot(theta,np.diag(l[B,:],0),axes=(3,0))

    # Apply U
    theta = np.tensordot(theta,U,axes=([1,2],[0,1]));

    # SVD
    theta = np.reshape(np.transpose(theta,(2,0,3,1)),(d*chi,d*chi));
    X, Y, Z = np.linalg.svd(theta); Z = Z.T

    # Truncate
    l[A,0:chi]=Y[0:chi]/np.sqrt(sum(Y[0:chi]**2))

    X=np.reshape(X[0:d*chi,0:chi],(d,chi,chi))
    G[A,:,:,:]=np.transpose(np.tensordot(np.diag(l[B,:]**(-1)),X,axes=(1,1)),(1,0,2));

    Z=np.transpose(np.reshape(Z[0:d*chi,0:chi],(d,chi,chi)),(0,2,1))
    G[B,:,:,:]=np.tensordot(Z,np.diag(l[B,:]**(-1)),axes=(2,0));

print "E_iTEBD =", -np.log(np.sum(theta**2))/delta/2, "\nE_exact = -1.063544409973365"
```



Tutorial: (B) TEBD

I) Check the convergence of the code for different parameters (number of iterations, bond dimension, and time steps).

II) Evaluate the phase diagram of the transverse field Ising model by plotting the magnetization, correlation length, and entanglement entropy.

III) Check the stability of the transition by adding new terms to the Hamiltonian: $\sum_i \sigma_i^x \sigma_{i+1}^x$ and $\sum_i \sigma_i^z$

Efficient simulation of interacting topological matter

I) Entanglement and Matrix-Product States

II) Time Evolving Block Decimation

III) Density-Matrix Renormalization Group

IV) Extracting topological invariants

Density matrix renormalization group (DMRG)

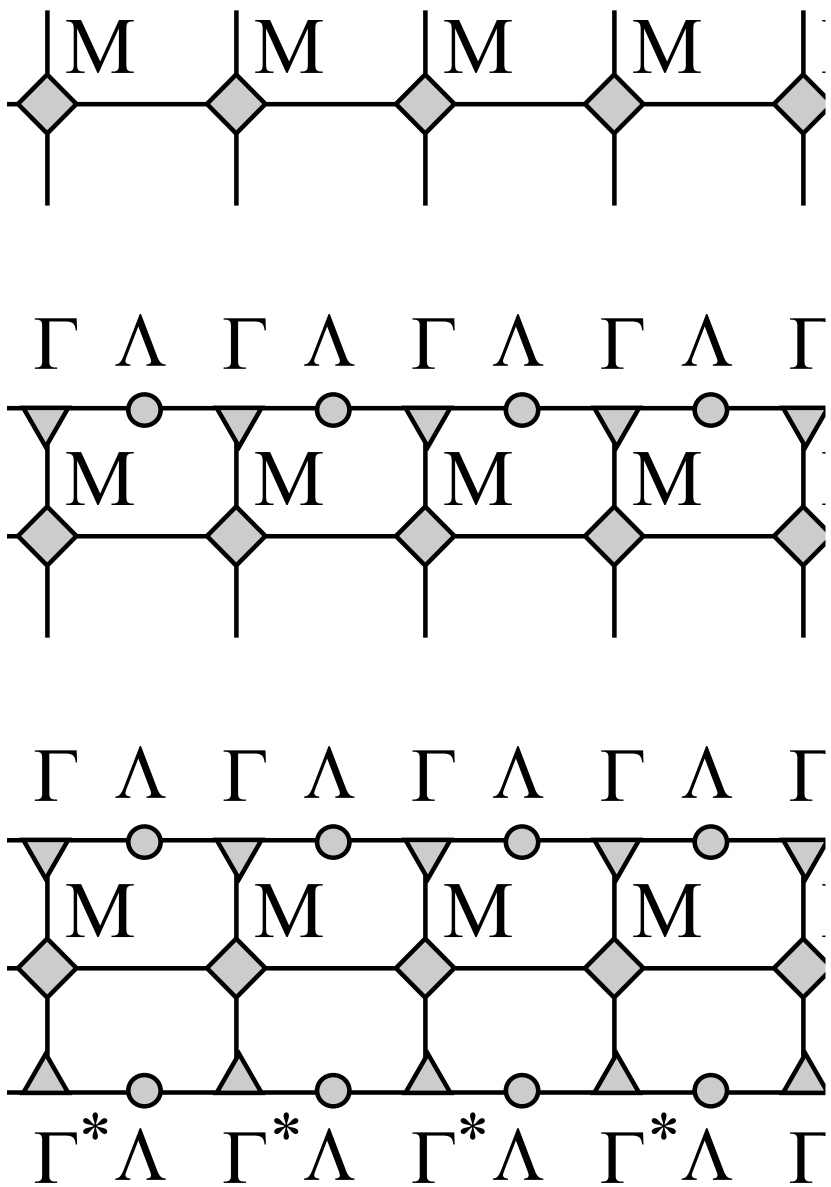
DMRG: Find the MPS representation of the ground state of a given Hamiltonian variationally
[White '92]

- Much faster convergence than TEBD
- Long range interactions can be easily implemented
- Similarly to TEBD, it allows to work directly in the thermodynamic limit

Density matrix renormalization group (DMRG)

Matrix-Product operators: Generalization of MPS to the space of linear operators

$$\mathcal{O}_{i_1 i_2 \ldots i_L,\ i'_1, i'_2 \ldots i'_L} =$$



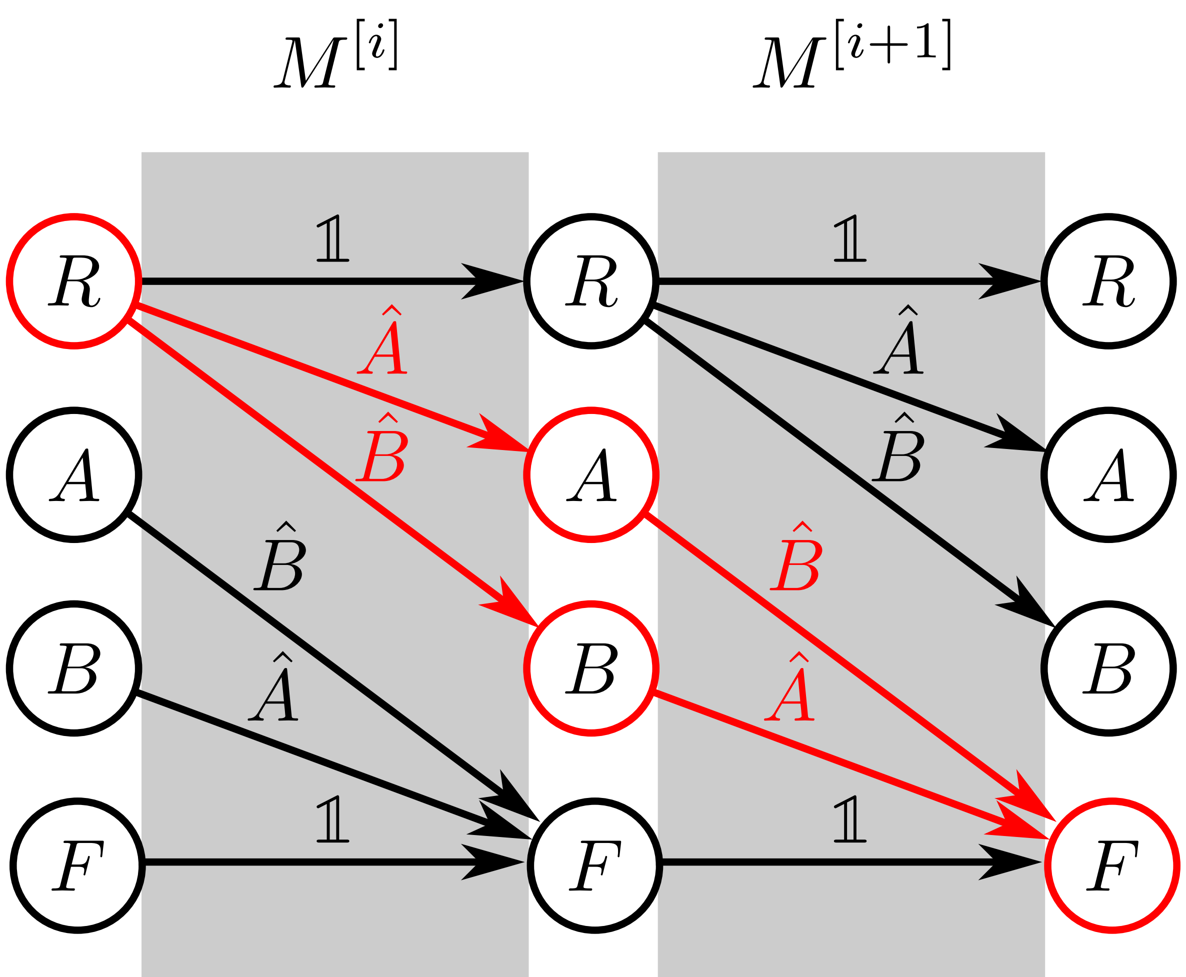
Density matrix renormalization group (DMRG)

MPO representation of $\hat{O} = \sum_i \left(\hat{A}_i \hat{B}_{i+1} + \hat{B}_i \hat{A}_{i+1} \right)$

$$
\begin{aligned}
&= \hat{A} \otimes \hat{B} \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1} \\
&\quad + \mathbb{1} \otimes \hat{A} \otimes \hat{B} \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1} + \cdots \\
&\quad + \hat{B} \otimes \hat{A} \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1} \\
&\quad + \mathbb{1} \otimes \hat{B} \otimes \hat{A} \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1} + \cdots
\end{aligned}
$$

Density matrix renormalization group (DMRG)

MPO representation of $\hat{O} = \sum_i \left(\hat{A}_i \hat{B}_{i+1} + \hat{B}_i \hat{A}_{i+1} \right)$

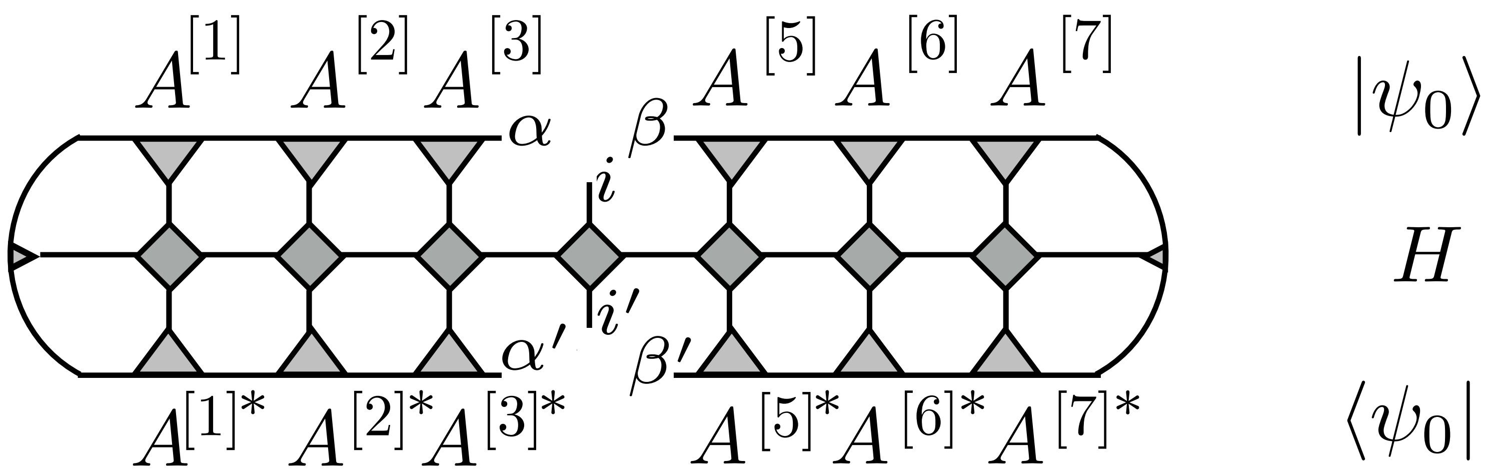


$$M = \begin{array}{cc} \begin{array}{cccc} R & A & B & F \end{array} & \\ \begin{pmatrix} \mathbb{1} & \hat{A} & \hat{B} & 0 \\ 0 & 0 & 0 & \hat{B} \\ 0 & 0 & 0 & \hat{A} \\ 0 & 0 & 0 & \mathbb{1} \end{pmatrix} & \begin{array}{c} R \\ A \\ B \\ F \end{array} \end{array}$$

Density matrix renormalization group (DMRG)

Find the **ground state** iteratively

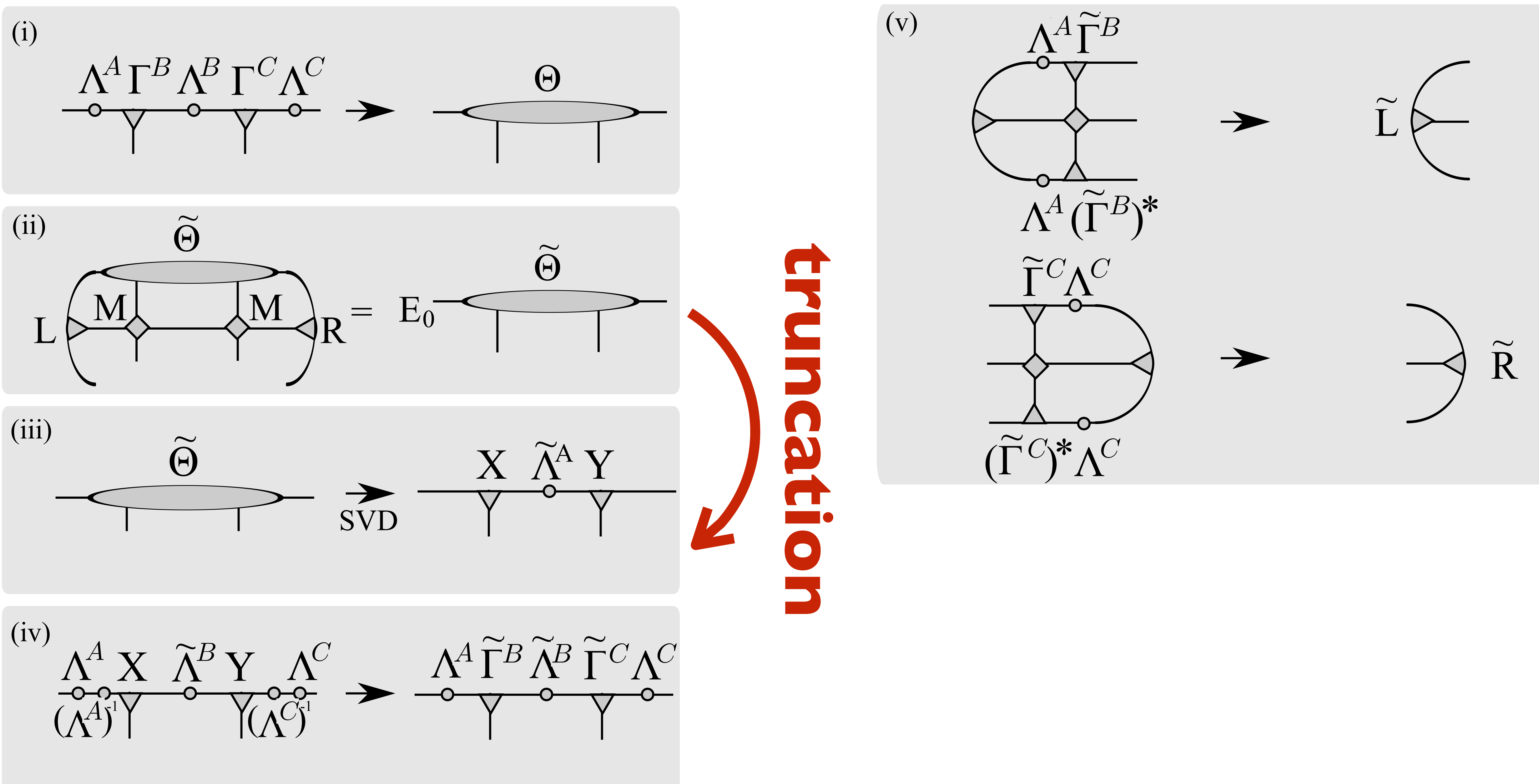
$H_{\alpha i\beta;\alpha' i'\beta'} =$



by locally minimizing energy of $H_{\alpha i\beta;\alpha' i'\beta'}$ (e.g., Lanczos)

Much faster convergence than TEBD + allows for long range interactions!

Density matrix renormalization group (DMRG)



Tensor Network Python (TeNPy)

What is TeNPy

- Python 3 library for simulations with tensor network
  https://github.com/tenpy/tenpy
- Object oriented, modular structure, and easy to install
- HTML documentation
  https://tenpy.github.io
- (in)finite DMRG, TEBD, TDVP

Tensor Network Python (TeNPy)

Example: DMRG

Example

```python
from tenpy.networks.mps import MPS
from tenpy.models.tf_ising import TFIChain
from tenpy.algorithms import dmrg

M = TFIChain({'L': 16, 'J': 1., 'g': 1.5})
psi = MPS.from_product_state(M.lat.mps_sites(),
                             ['up']*16, 'finite')
dmrg_params = {'trunc_params': {'chi_max': 30,
                                'svd_min': 1.e-10}}
dmrg.run(psi, M, dmrg_params) # find ground state
print("E =", sum(M.bond_energies(psi)))
print("final bond dimensions: ", psi.chi)
```

Tutorial: (C) TeNPy DMRG

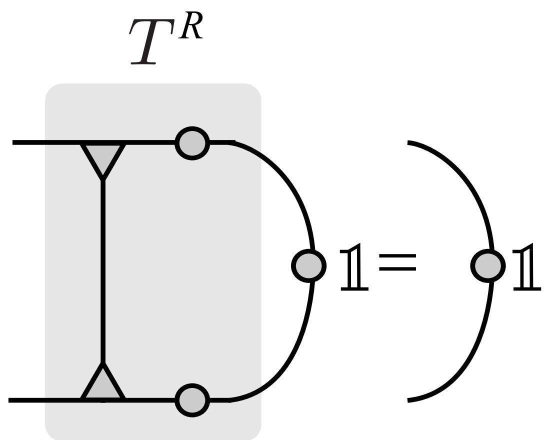
I) Get familiar with the TeNPy interface: https://tenpy.github.io

II) Extract the central charge using "entanglement scaling": $S = \frac{c}{6} \log \xi + \text{const}$

Correlation length of infinite systems

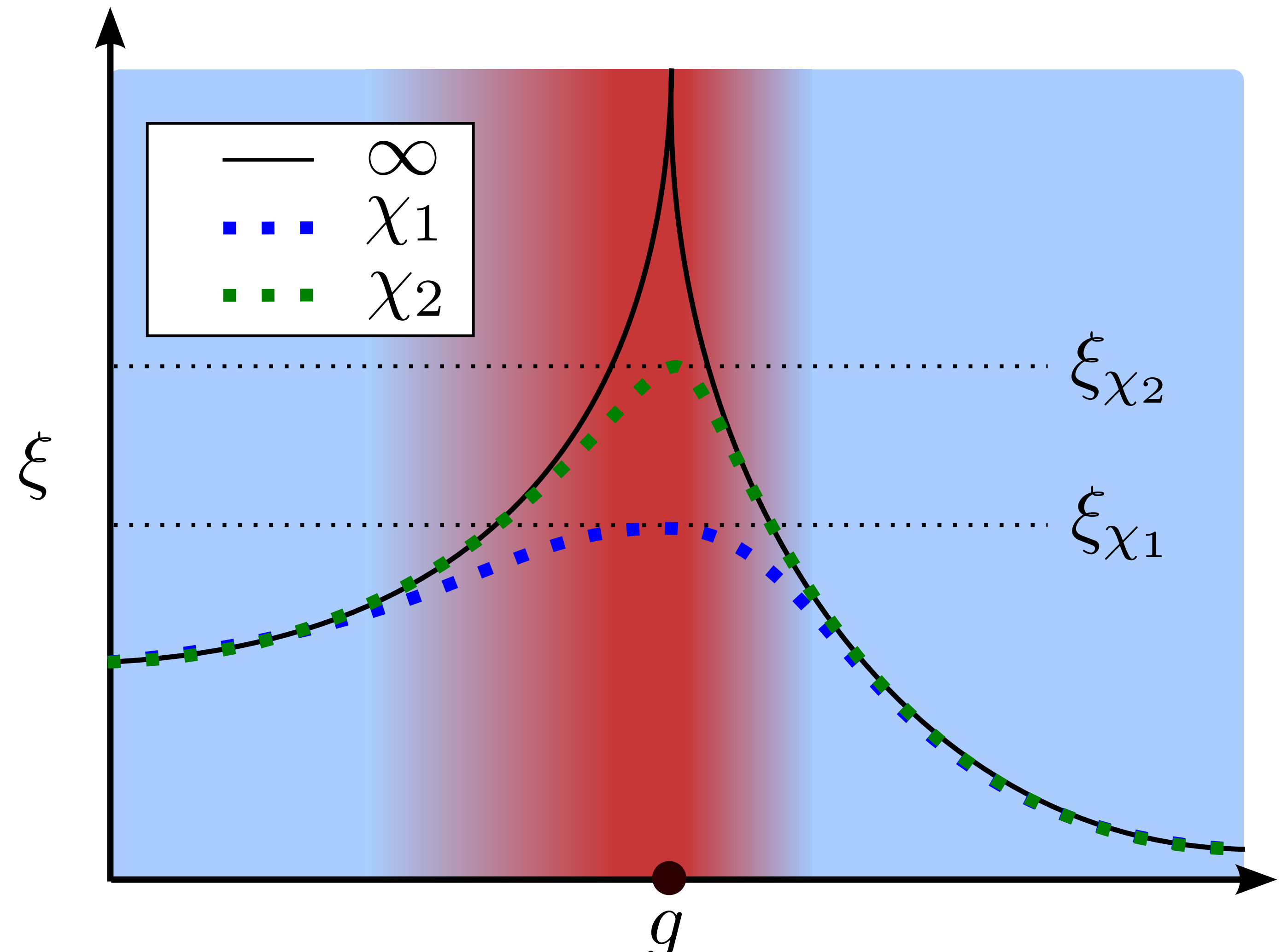
Correlation length of an MPS:

$$\xi = -\frac{1}{\log|\epsilon_2|},$$



ϵ_2 is the second largest eigenvalue of the transfer matrix

Finite entanglement scaling: Entanglement and correlation length are always finite in a finite dimensional MPS



Efficient simulation of interacting topological matter

I) Entanglement and Matrix-Product States

II) Time Evolving Block Decimation

III) Density-Matrix Renormalization Group

IV) Extracting topological invariants

Topological Phases of Matter

"Non invertible"

-FQHE
-Toric Code
-Spin Liquids

Anyons

"SET"

"Symmetry protected top."

-Topological Insulators
-Haldane Phase

Symmetry Fractionalization

"Symmetry broken"

"Chiral top."

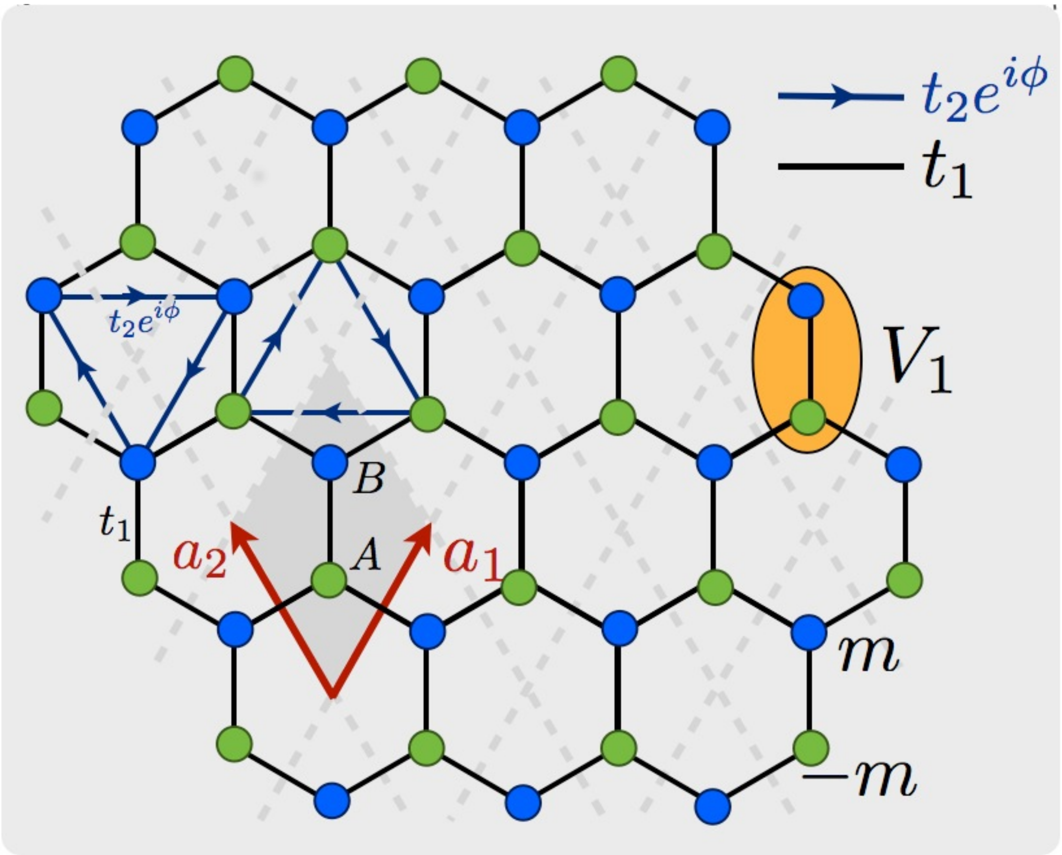
-IQHE
-Chern Insulator

Chiral Edge Modes

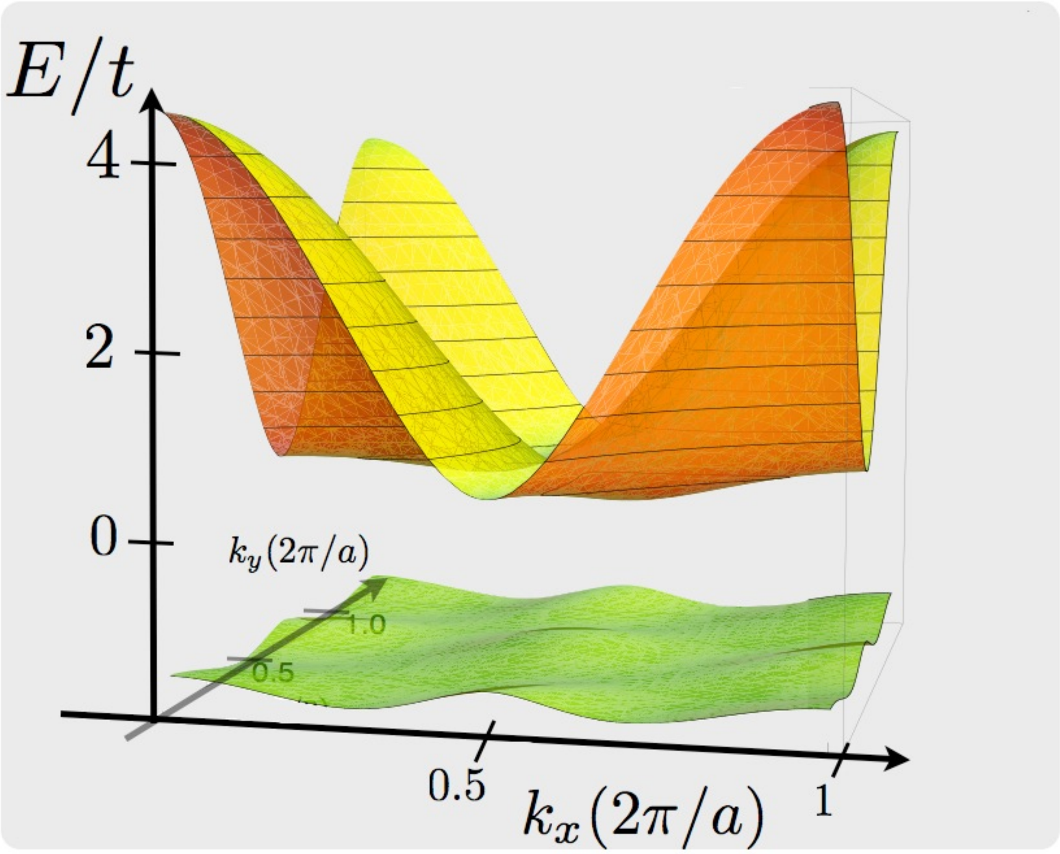
"Trivially disordered"

All phases can be identified using DMRG!

(Fractional) Chern Insulators



[Haldane '88]



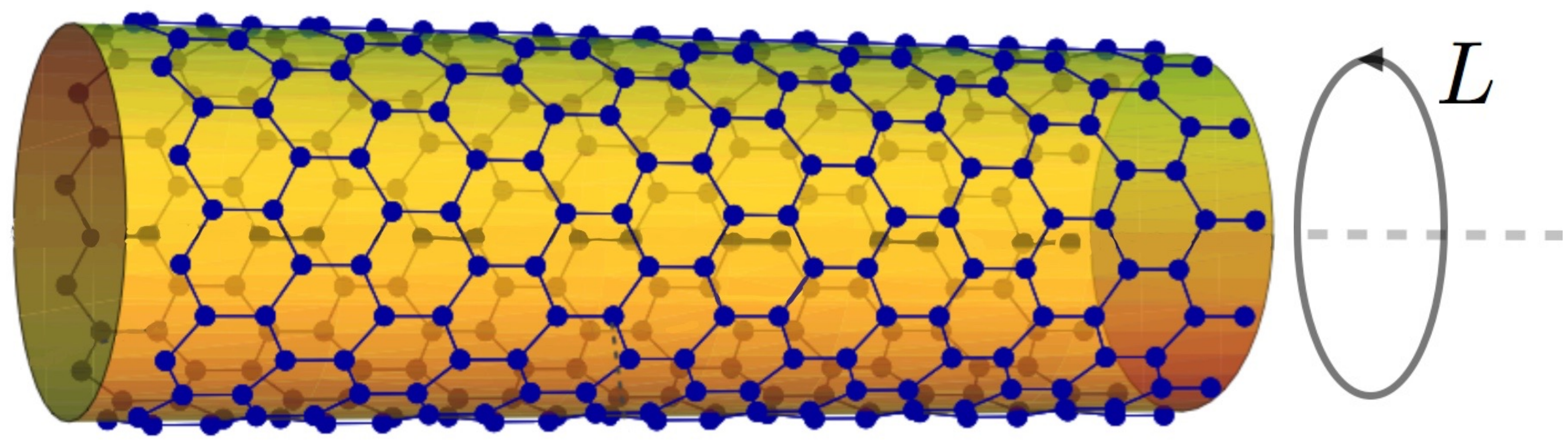
D. Sheng, Z.-C. Gu, K. Sun, and L. Shen '11
N. Regnault and B. A. Bernevig '11
E. J. Bergholtz and Z. Liu '13
S. Kourtis, T. Neupert, C. Chamon, and C. Mudry, '12
S. Kourtis, J. W. F. Venderbos, and M. Daghofer '13
....

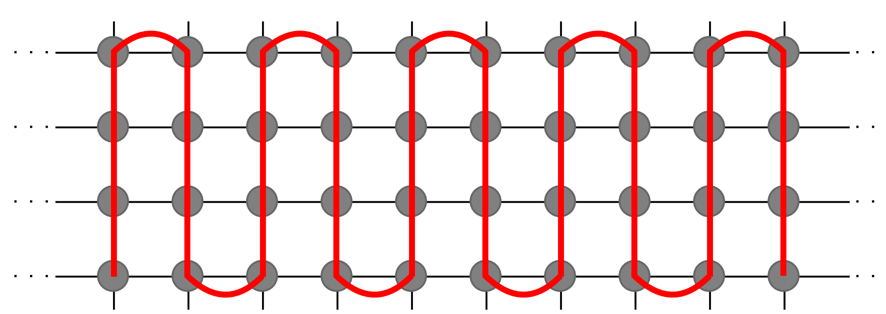
Chern Insulator ($\nu = 1$)

Fractional Chern Insulator (e.g., $\nu = 1/3$):
Interactions are important!

DMRG for 2D Systems

DMRG on cylinders with circumference up to $L = 12$



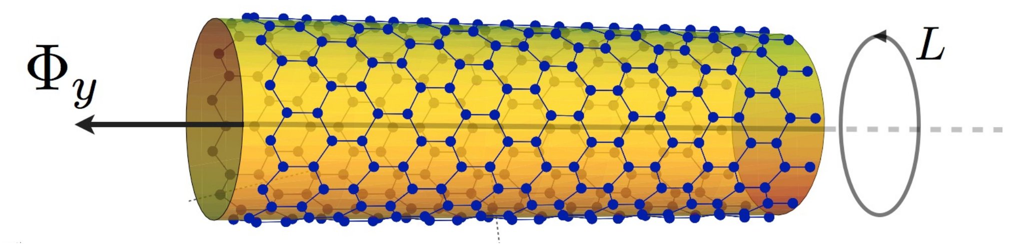


2D physics at cost of long range interaction in 1D representation!

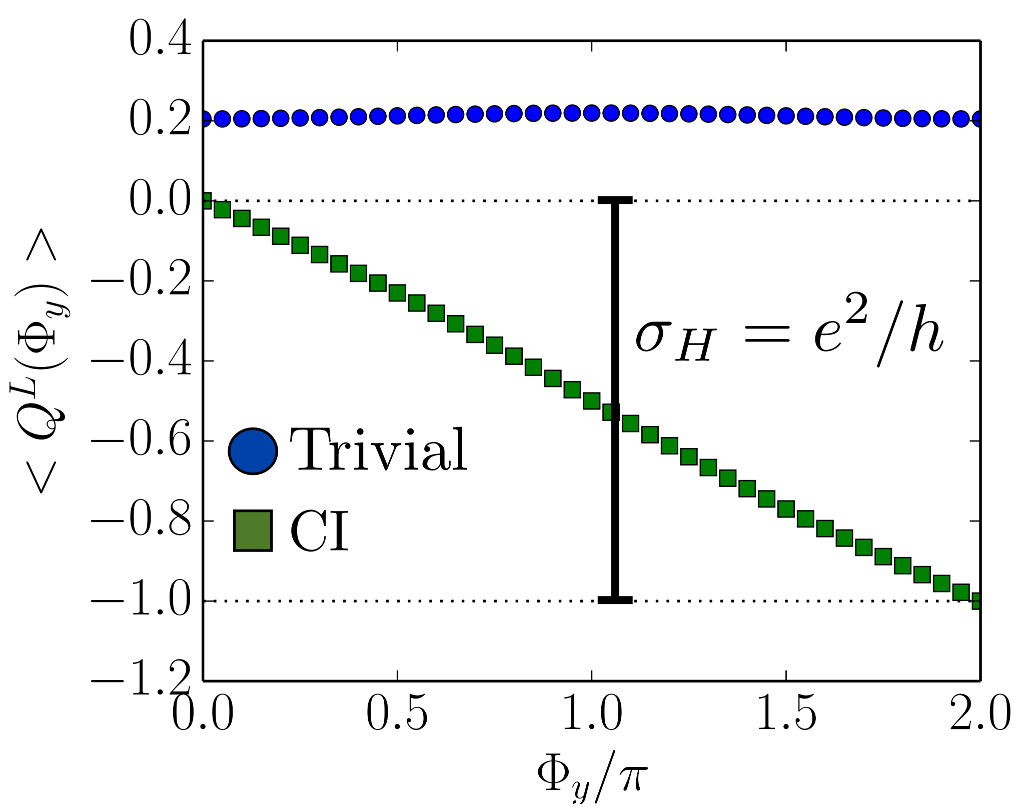
- MPO representation of the Hamiltonian (bond dimension scales **polynomially** with L)
- Area law: MPS dimensions of the ground state grows **exponentially** with L!

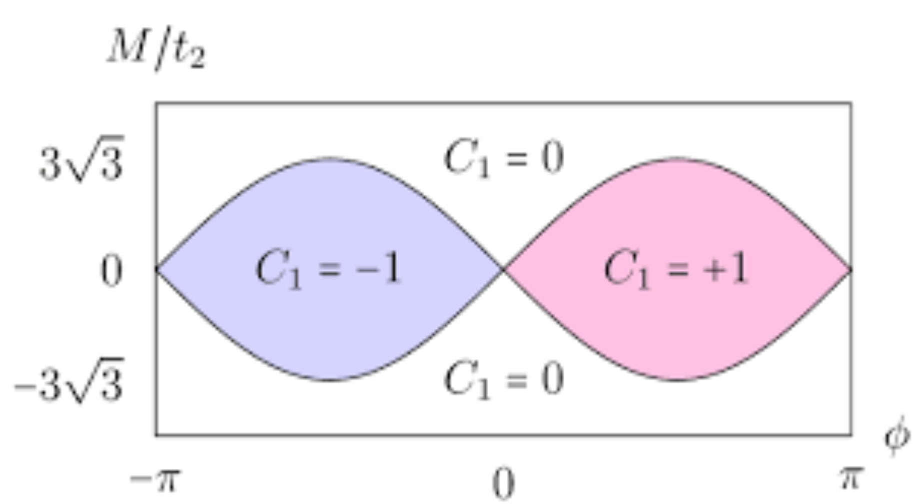
Chern Insulators

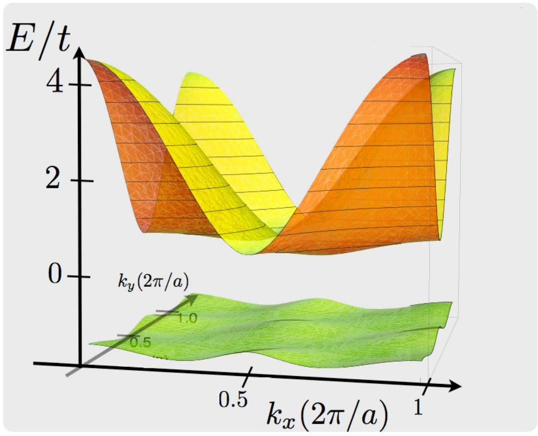
Laughlin charge pump



- Generate flux ϕ_y using complex hopping at the boundary
- Start from $\phi_y = 0$ and adiabatically insert a flux 2π
- Measure the charge pumped from left to right

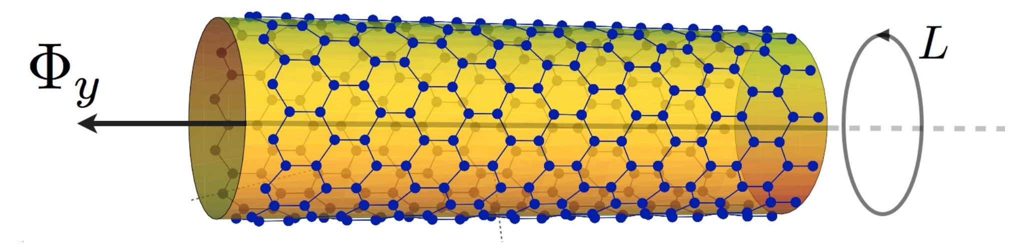


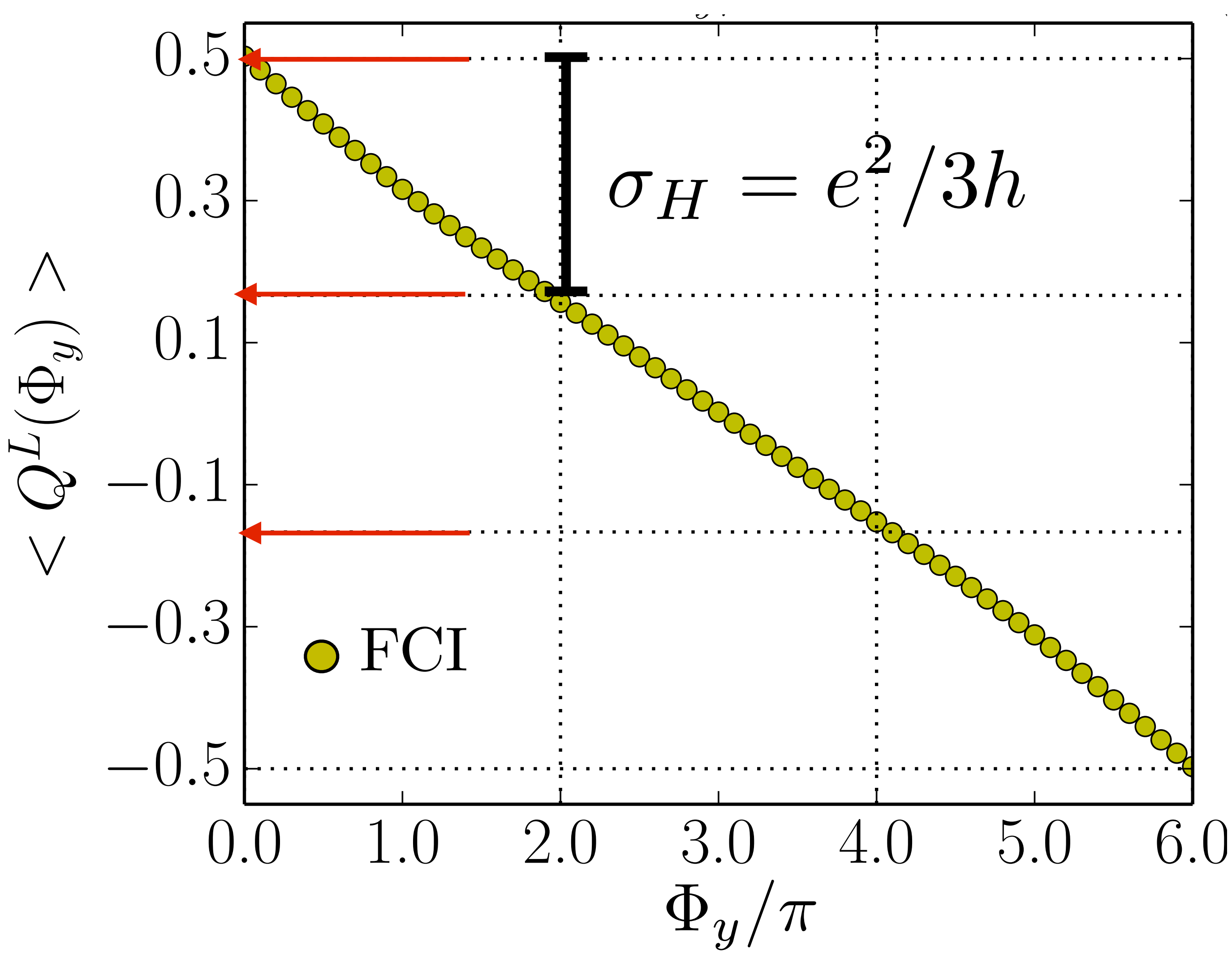


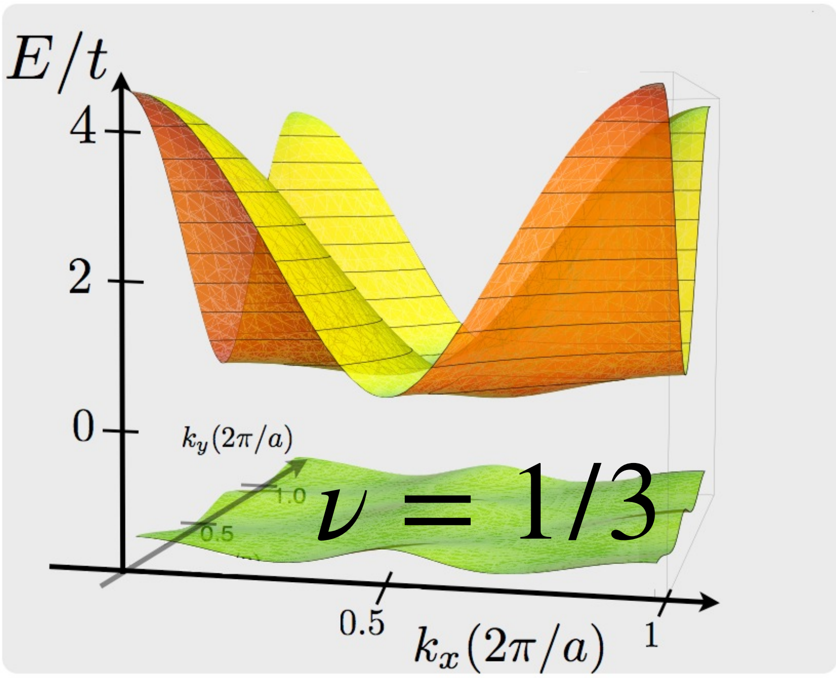


Fractional Chern Insulators

Laughlin charge pump



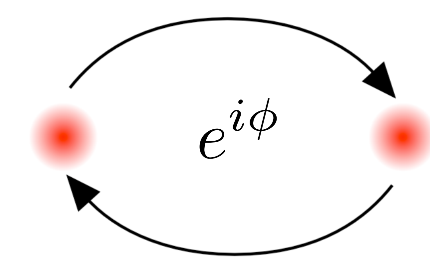




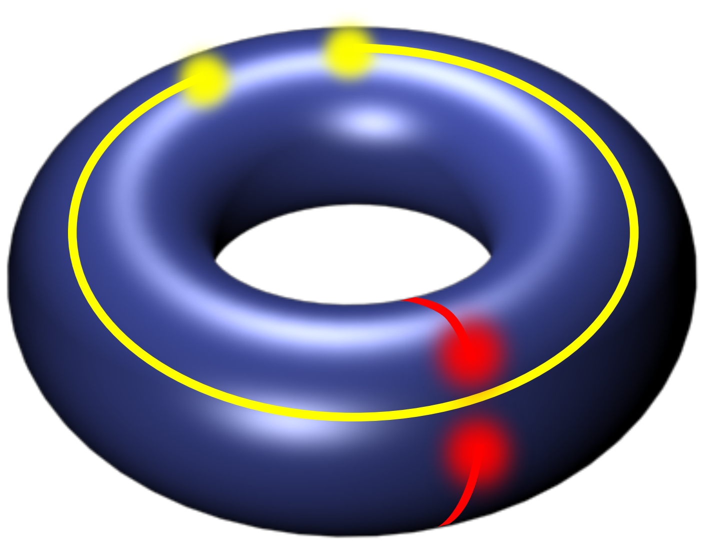
Fractional Chern Insulators: Top. Entanglement

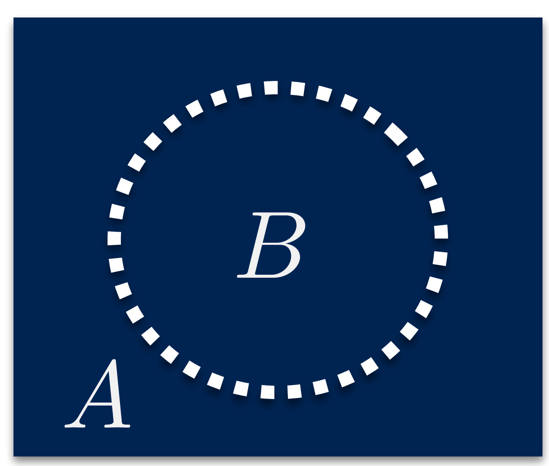
Intrinsic topological order: Gapped quantum phases that are robust to any small (local) perturbation

Characterized by quasiparticle excitations that obey **fractional statistics "anyons"** [Wen '90]



- Topological degeneracy on torus/cylinder (= number of anyons)
- Topological entanglement entropy γ : $S = \alpha L - \gamma$
  [Kitaev and Preskill '06, Levin and Wen '06]





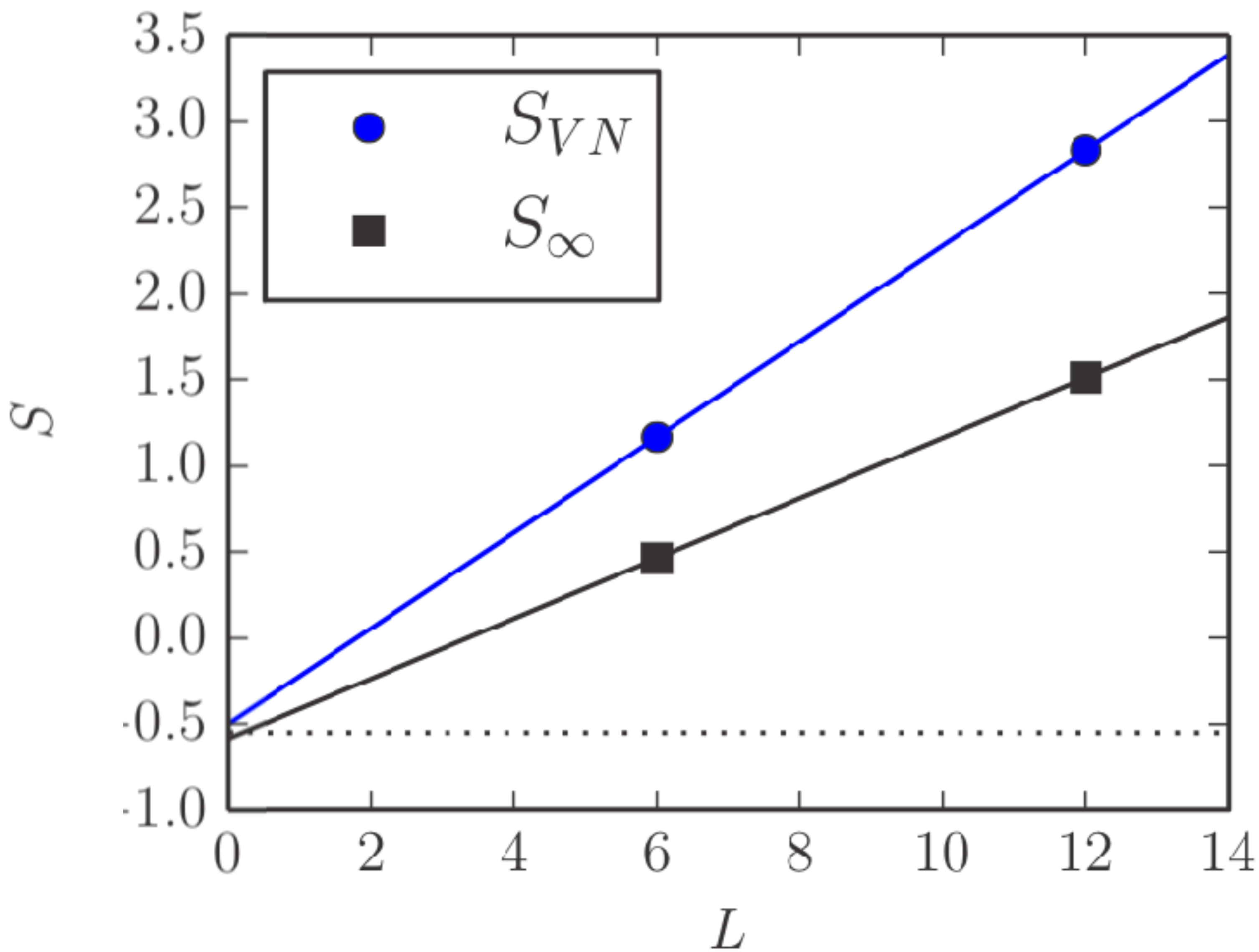
$$|\psi\rangle = \sum_\alpha \sqrt{p_\alpha} \quad |\phi_\alpha^A\rangle \quad |\phi_\alpha^B\rangle$$

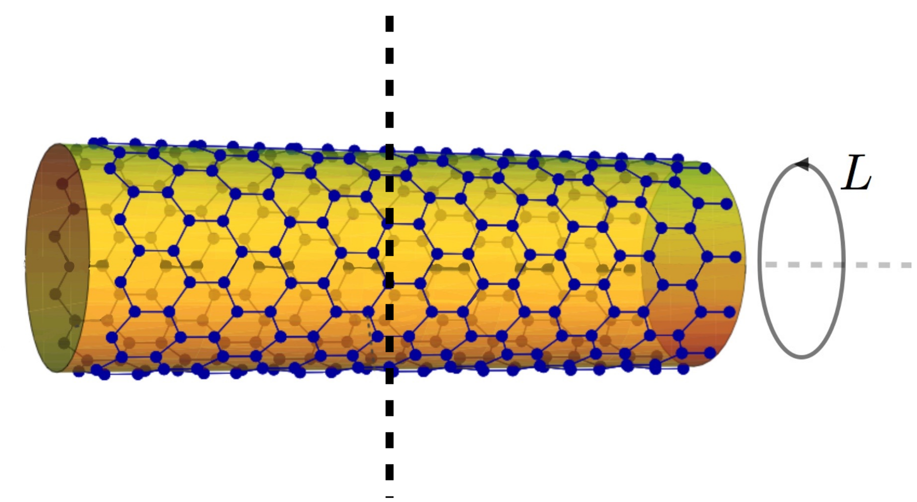
$$S = -\sum_\alpha p_\alpha \log p_\alpha$$

Abelian: $\gamma = \log(\sqrt{\#\mathrm{anyons}})$

Fractional Chern Insulators: Top. Entanglement

Topological entanglement in the FCI phase



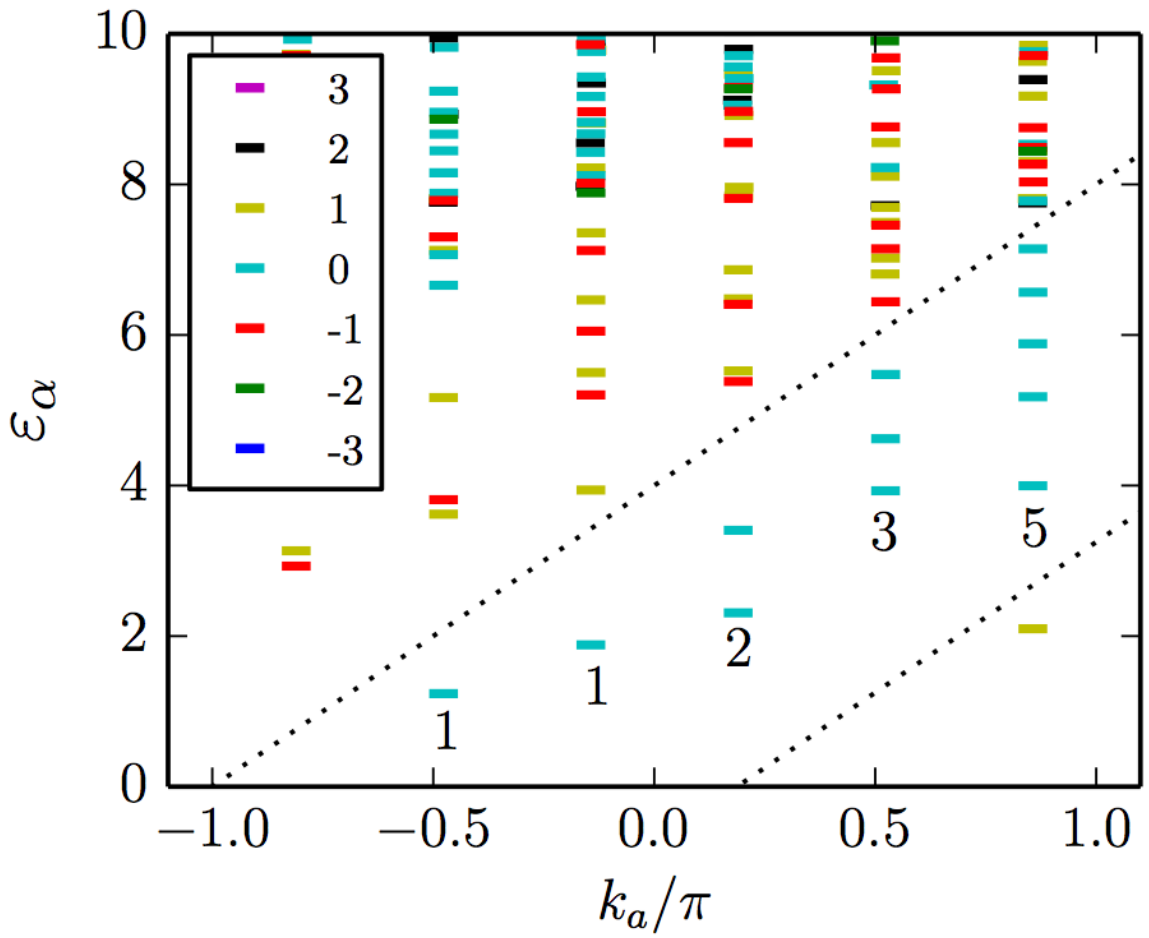


$$S = \alpha L - \gamma$$

$$\log \sqrt{3}$$

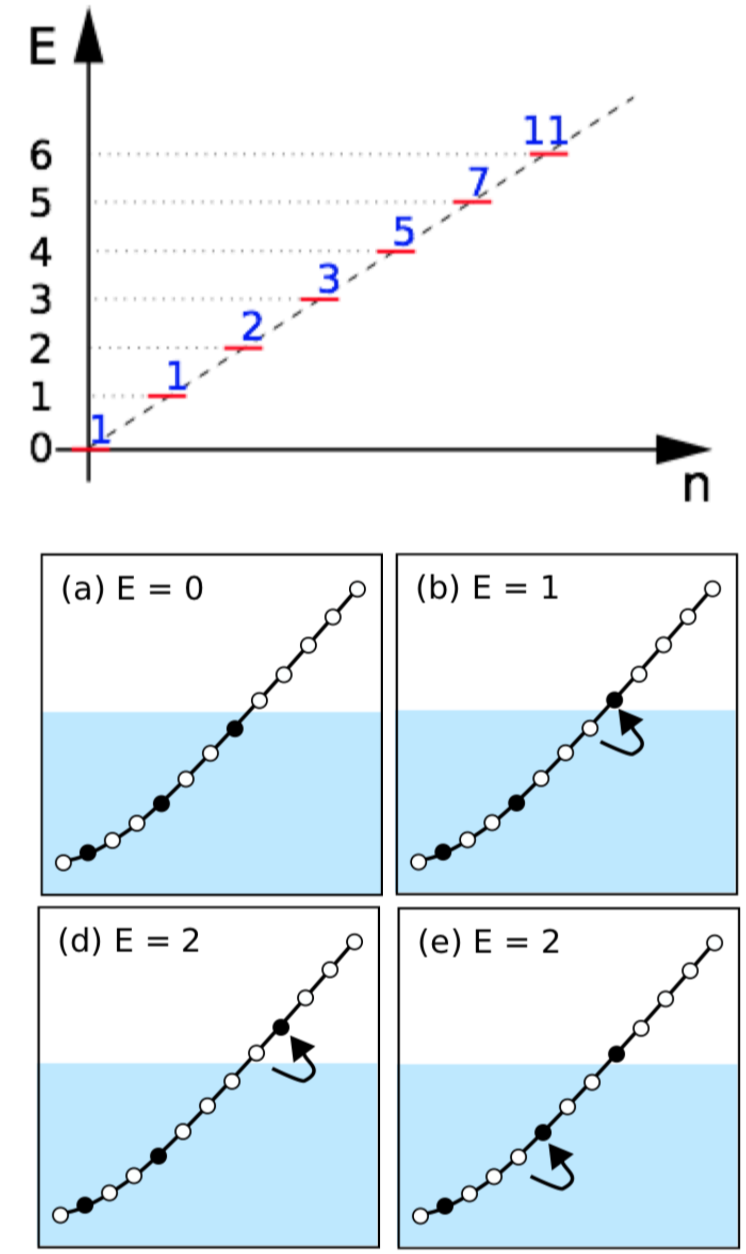
Fractional Chern Insulators: Chiral Edge States

Chiral edges states



[Li & Haldane '08]

$$\rho^{\mathrm{red}} = \sum_\alpha \exp(-\epsilon_\alpha)|\phi_\alpha\rangle\langle\phi_\alpha|$$

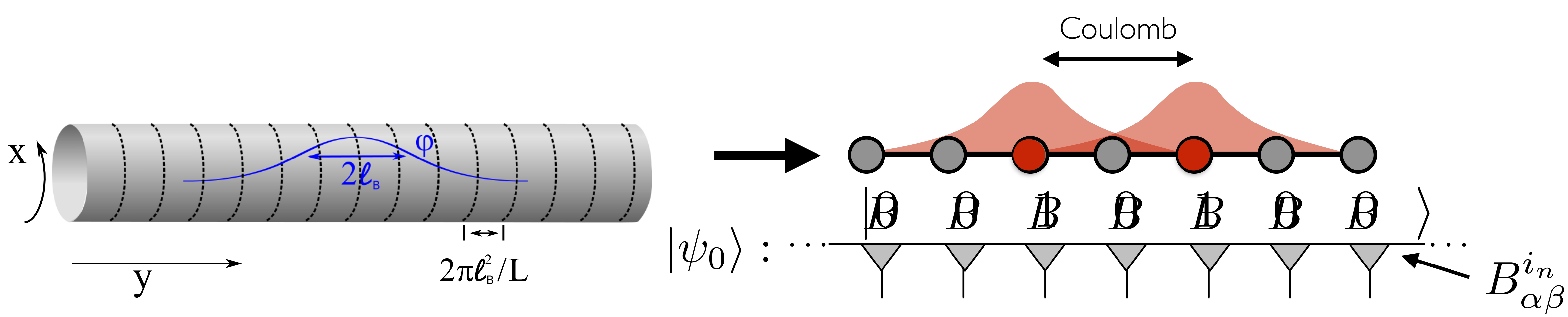


Fractional Quantum Hall

Density Matrix Renormalization Group
to simulate FQHE on infinite cylinders

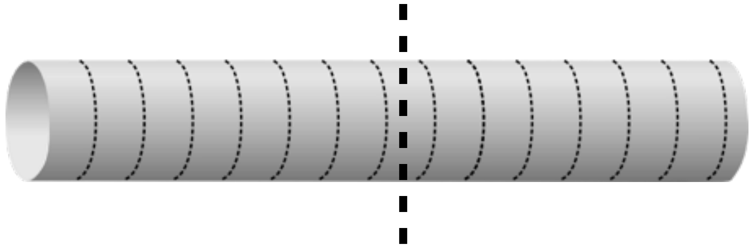
Orbitals in first Landau level are localized along the cylinder: **Quasi 1D model**

[Tao & Taoless 88, Haldane & Rezayi '94; Bergholtz et al. '05, Seidel et al. '05]

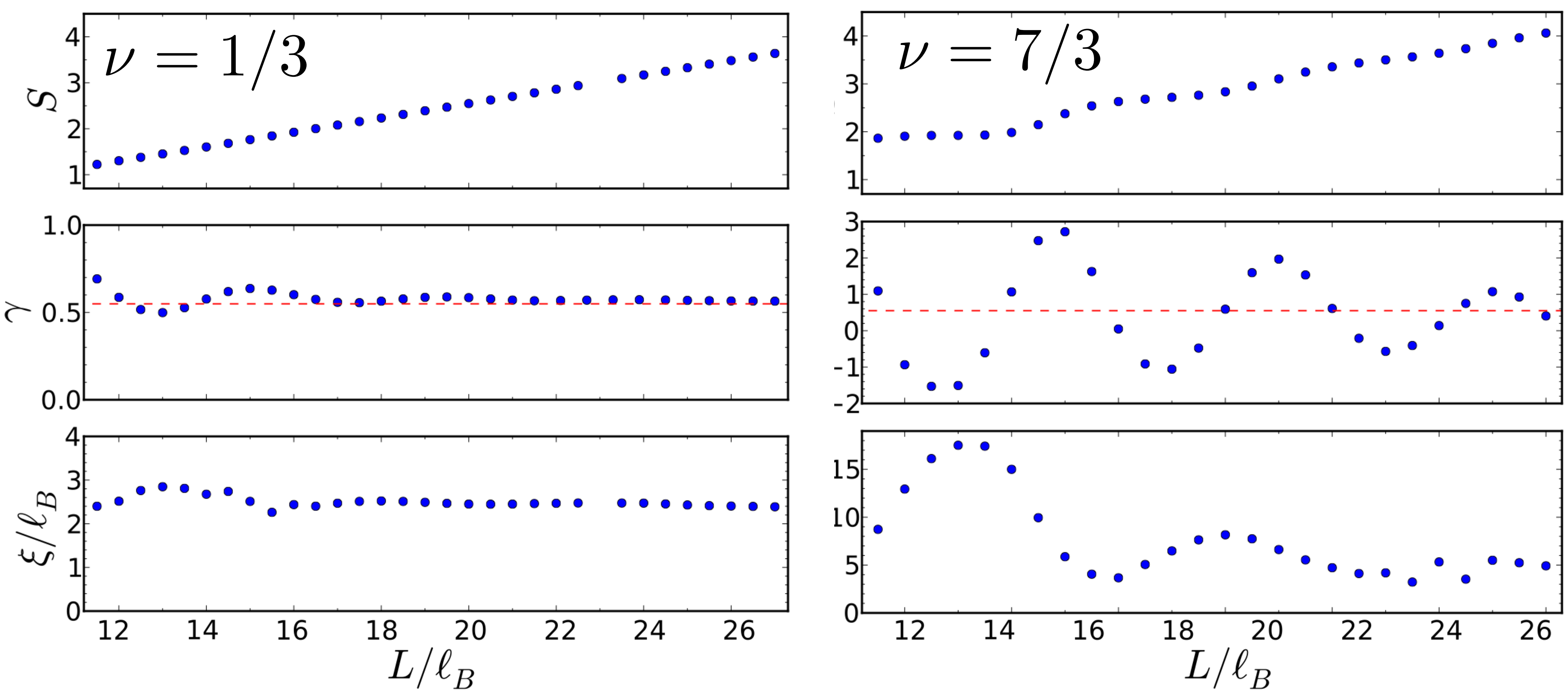


Fractional Quantum Hall

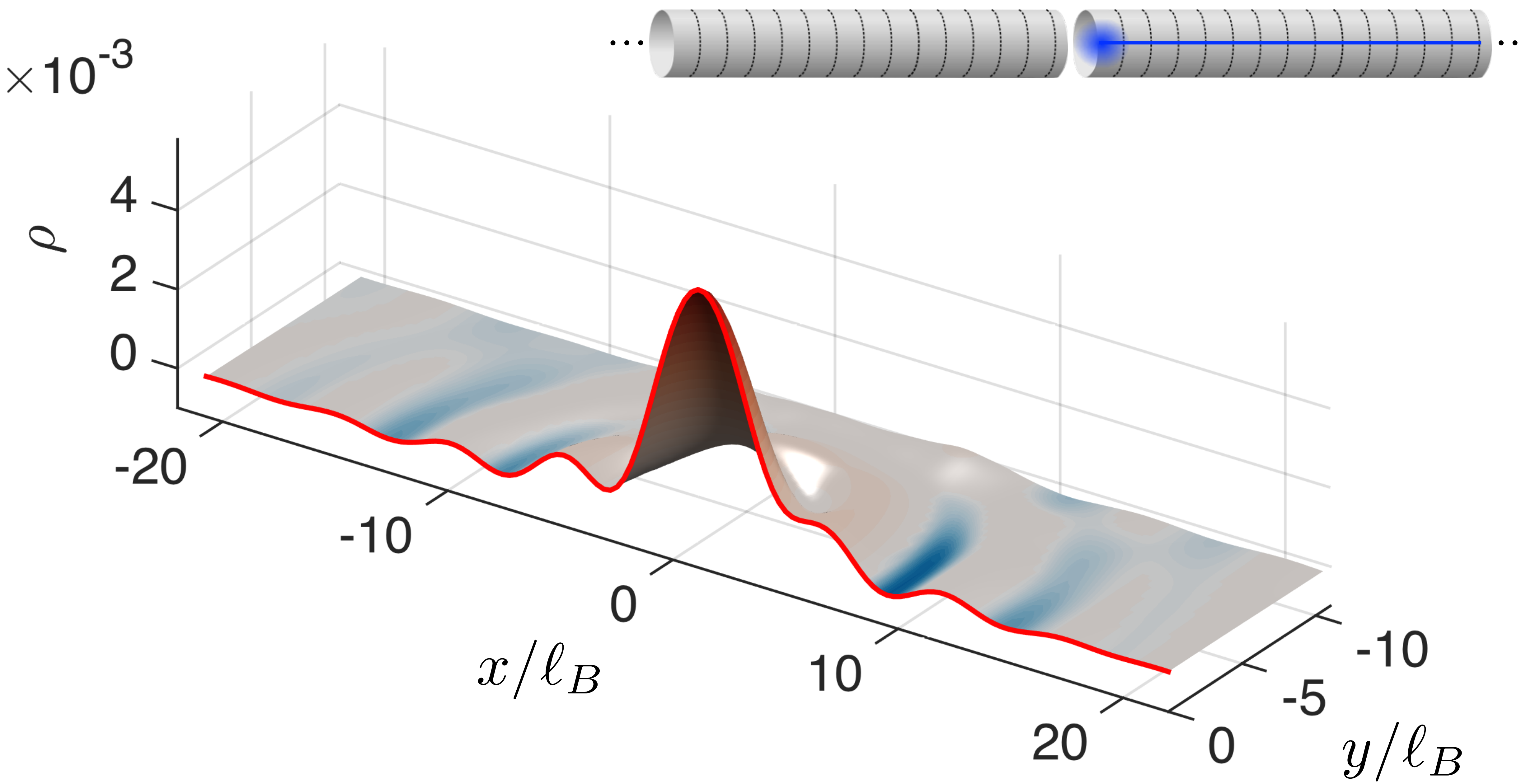
Topological entanglement entropy of the FQHE with Coulomb interactions ("minimally entangled states")



$$S = \alpha L - \gamma$$



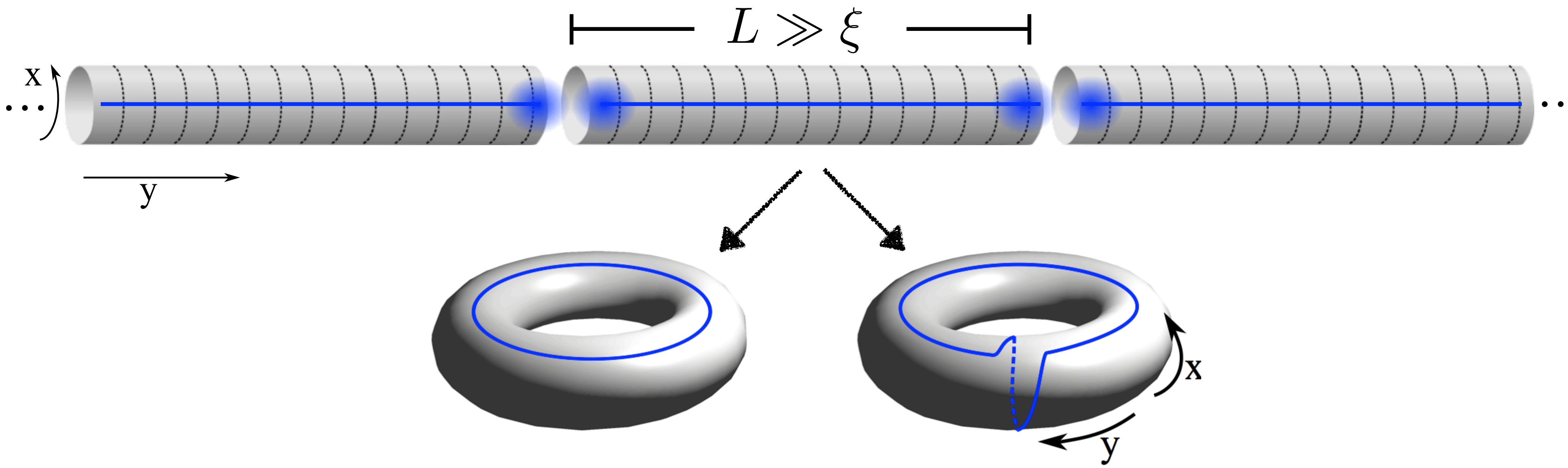
Fractional Quantum Hall



Density profile for Fibonacci anyons with charge e/5

Fractional Quantum Hall

Extracting topological content by adding a "twist"



Momentum polarization: **topological spin, central charge, Hall viscosity**

$$U_{T;ab} = \delta_{ab} \exp\left[2\pi i \left(h_a - \frac{c}{24} - \frac{\eta_H}{2\pi\hbar} L_x^2\right)\right]$$

Tutorial: (D) TeNPy Entanglement Spectrum

I) Calculate the entanglement spectrum of the Haldane model. Can you get the counting shown in Steve's lecture?

II) Calculate the expectation value of the charge density operator and explore the regime of large interactions.

Tutorial: (E) TeNPy Laughlin Pump

I) Explore a few points in the phase diagram of the Haldane using the Laughlin pump.

II) Open ended: Can you find parameters to stabilize an FCI?

Efficient simulation of interacting topological matter

I) Entanglement and Matrix-Product States

II) Time Evolving Block Decimation

III) Density-Matrix Renormalization Group

IV) Extracting topological invariants

Python toolbox for MPS/TPS simulations:

https://tenpy.github.io

Lecture notes on MPS:
Hauschild and FP, arxiv:1805.00055

Thank You!