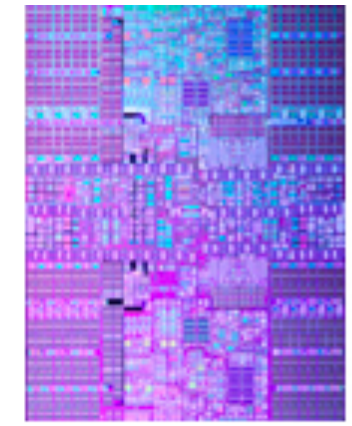
$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$



Introduction to Exact Diagonalization

Andreas Läuchli,
"New states of quantum matter"
MPI für Physik komplexer Systeme - Dresden

http://www.pks.mpg.de/~aml
laeuchli@comp-phys.org

IPAM QS2009 Workshop, January 23, 2009

Outline

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

- Main Idea
- Examples of typical applications
- Structure of an Exact Diagonalization package
  - Hilbertspace, Symmetries
  - Hamiltonian
  - Linear Algebra
  - Observables
- Parallelization
- Summary & Outlook

Exact Diagonalization: Main Idea

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

Exact Diagonalization: Main Idea

- Solve the Schrödinger equation of a quantum many body system numerically

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

Exact Diagonalization: Main Idea

- Solve the Schrödinger equation of a quantum many body system numerically

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

- Sparse matrix, but for quantum many body systems the vector space dimension grows exponentially!

Exact Diagonalization: Main Idea

- Solve the Schrödinger equation of a quantum many body system numerically

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

- Sparse matrix, but for quantum many body systems the vector space dimension grows exponentially!
- Some people will tell you that's all there is.

Exact Diagonalization: Main Idea

- Solve the Schrödinger equation of a quantum many body system numerically

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

- Sparse matrix, but for quantum many body systems the vector space dimension grows exponentially!

- Some people will tell you that's all there is.

- But if you want to get a maximum of physical information out of a finite system there is a lot more to do and the reward is a powerful:

Exact Diagonalization: Main Idea

- Solve the Schrödinger equation of a quantum many body system numerically

$$\mathcal{H}|\psi\rangle = E|\psi\rangle$$

- Sparse matrix, but for quantum many body systems the vector space dimension grows exponentially!

- Some people will tell you that's all there is.

- But if you want to get a maximum of physical information out of a finite system there is a lot more to do and the reward is a powerful:

Quantum Mechanics Toolbox

Exact Diagonalization: Applications

Exact Diagonalization: Applications

- Quantum Magnets: nature of novel phases, critical points in 1D, dynamical correlation functions in 1D & 2D

Exact Diagonalization: Applications

- Quantum Magnets: nature of novel phases, critical points in 1D, dynamical correlation functions in 1D & 2D

- Fermionic models (Hubbard/t-J): gaps, pairing properties, correlation exponents, etc

Exact Diagonalization: Applications

- Quantum Magnets: nature of novel phases, critical points in 1D, dynamical correlation functions in 1D & 2D
- Fermionic models (Hubbard/t-J): gaps, pairing properties, correlation exponents, etc
- Fractional Quantum Hall states: energy gaps, overlap with model states, entanglement spectra

Exact Diagonalization: Applications

- Quantum Magnets: nature of novel phases, critical points in 1D, dynamical correlation functions in 1D & 2D

- Fermionic models (Hubbard/t-J): gaps, pairing properties, correlation exponents, etc

- Fractional Quantum Hall states: energy gaps, overlap with model states, entanglement spectra

- Quantum dimer models or other constrained models:
  → talk by D. Poilblanc on Tuesday afternoon

Exact Diagonalization: Applications

- Quantum Magnets: nature of novel phases, critical points in 1D, dynamical correlation functions in 1D & 2D
- Fermionic models (Hubbard/t-J): gaps, pairing properties, correlation exponents, etc
- Fractional Quantum Hall states: energy gaps, overlap with model states, entanglement spectra
- Quantum dimer models or other constrained models:
  → talk by D. Poilblanc on Tuesday afternoon
- Full Configuration Interaction in Quantum Chemistry

Exact Diagonalization: Present Day Limits

low-lying eigenvalues, not full diagonalization

Exact Diagonalization: Present Day Limits

- Fractional quantum hall effect
  different filling fractions ν, up to 16-20 electrons
  up to 300 million basis states, up to 1 billion in the near future

low-lying eigenvalues, not full diagonalization

Exact Diagonalization: Present Day Limits

- Fractional quantum hall effect
  different filling fractions ν, up to 16-20 electrons
  up to 300 million basis states, up to 1 billion in the near future

- Spin S=1/2 models:
  40 spins square lattice, 39 sites triangular, 42 sites star lattice at $S^z=0$
  64 spins or more in elevated magnetization sectors
  up to 1.5 billion(=10^9) basis states with symmetries, up to 4.5 billion without

low-lying eigenvalues, not full diagonalization

Exact Diagonalization: Present Day Limits

- Fractional quantum hall effect
  different filling fractions ν, up to 16-20 electrons
  up to 300 million basis states, up to 1 billion in the near future

- Spin S=1/2 models:
  40 spins square lattice, 39 sites triangular, 42 sites star lattice at $S^z=0$
  64 spins or more in elevated magnetization sectors
  up to 1.5 billion($=10^9$) basis states with symmetries, up to 4.5 billion without

- t-J models:
  32 sites checkerboard with 2 holes
  32 sites square lattice with 4 holes
  up to 2.8 billion basis states

low-lying eigenvalues, not full diagonalization

Exact Diagonalization: Present Day Limits

- Fractional quantum hall effect

  different filling fractions ν, up to 16-20 electrons

  up to 300 million basis states, up to 1 billion in the near future

- Spin S=1/2 models:

  40 spins square lattice, 39 sites triangular, 42 sites star lattice at $S^z=0$

  64 spins or more in elevated magnetization sectors

  up to 1.5 billion(=10^9) basis states with symmetries, up to 4.5 billion without

- t-J models:

  32 sites checkerboard with 2 holes

  32 sites square lattice with 4 holes

  up to 2.8 billion basis states

- Hubbard models

  20 sites square lattice at half filling, 20 sites quantum dot structure

  22-25 sites in ultracold atoms setting

  up to 80 billion basis states

low-lying eigenvalues, not full diagonalization

Structure of an Exact Diagonalization code

Ingredients

Ingredients

- Hilbert space

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization
  - Lanczos type diagonalization (needs only $|v\rangle = H|u\rangle$ operations)

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization
  - Lanczos type diagonalization (needs only $|v\rangle = H|u\rangle$ operations)
  - More exotic eigensolver techniques, real oder imaginary-time propagation,

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization
  - Lanczos type diagonalization (needs only $|v\rangle = H|u\rangle$ operations)
  - More exotic eigensolver techniques, real oder imaginary-time propagation,
- Observables

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization
  - Lanczos type diagonalization (needs only $|v\rangle = H|u\rangle$ operations)
  - More exotic eigensolver techniques, real oder imaginary-time propagation,
- Observables
  - Static quantities (multipoint correlation functions, correlation density matrices,...)

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization
  - Lanczos type diagonalization (needs only $|v\rangle = H|u\rangle$ operations)
  - More exotic eigensolver techniques, real oder imaginary-time propagation,
- Observables
  - Static quantities (multipoint correlation functions, correlation density matrices,...)
  - Dynamic observables (spectral functions, density of states,...)

Ingredients

- Hilbert space
  - Basis represention, Lookup techniques
  - Symmetries
- Hamiltonian Matrix
  - Sparse Matrix representation (memory/disk)
  - Matrix recalculation on the fly (matrix-free)
- Linear Algebra : Eigensolver / Time propagation
  - LAPACK full diagonalization
  - Lanczos type diagonalization (needs only $|v\rangle = H|u\rangle$ operations)
  - More exotic eigensolver techniques, real oder imaginary-time propagation,
- Observables
  - Static quantities (multipoint correlation functions, correlation density matrices,...)
  - Dynamic observables (spectral functions, density of states,...)
  - Real-time evolution

Hilbert Space

Basis representation

- States of the Hilbert space need to be represented in the computer.
- Choose a representation which makes it simple to act with the Hamiltonian or other operators on the states, and to localize a given state in the basis
- Simple example: ensemble of S=1/2 sites in binary coding

$$|\uparrow\uparrow\downarrow\uparrow\rangle \rightarrow [1\ 1\ 0\ 1]_2 = 13$$

detection of up or down spin can be done with bit-test.
transverse exchange $S^+S^- + S^-S^+$ can be performed by an XOR operation:

$$\underbrace{[1\ 1\ 0\ 1]_2}_{\text{initial config}}\ \mathrm{XOR}\ \underbrace{[0\ 1\ 1\ 0]_2}_{\text{bit 1 at the two sites coupled}} = \underbrace{[1\ 0\ 1\ 1]_2}_{\text{final config}}$$

- For S=1, one bit is obviously not sufficent. Use ternary representation or simply occupy two bits to label the 3 states.

Basis representation

- For t-J models at low doping it is useful to factorize hole positions and spin configurations on the occupied sites.
- For Hubbard models one can factorize the Hilbert space in up and down electron configurations.
- For constrained models - such as dimer models - the efficient generation of all basis states requires some thought.

- One of the key challenges for a fast ED code is to find the index of the new configuration in the list of all configurations (index f in $H_{i,f}$).
- Let us look at the example of S=1/2 spins at fixed S^z

Basis lookup procedures (Lin tables)

- One of the key problems for a fast ED code is to find the index of the new configuration in the list of all configurations (index f in $H_{i,f}$).

$$[1\ 0\ 1\ 1]_2 = 11_{10}$$

- But is 11 the index of this configuration in a list of all S^z=3/2 states ? no !

- Use Lin tables to map from binary number to index in list of allowed states:
(generalization of this idea works for arbitrary number of additive quantum numbers)

- Two tables with $2^{(N/2)}$ [=sqrt(2^N)] entries, one for MSBs and one for LSBs

$$\begin{aligned}
[0\ 0] &= X \\
[0\ 1] &= 0 \\
[1\ 0] &= 1 \\
[1\ 1] &= 2
\end{aligned}$$

MSB

$$\begin{aligned}
[0\ 0] &= X \\
[0\ 1] &= 0 \\
[1\ 0] &= 1 \\
[1\ 1] &= 0
\end{aligned}$$

LSB

$$\begin{aligned}
\mathrm{Ind}([0\ 1\ 1\ 1]) &= 0 + 0 &= 0 \\
\mathrm{Ind}([1\ 0\ 1\ 1]) &= 1 + 0 &= 1 \\
\mathrm{Ind}([1\ 1\ 0\ 1]) &= 2 + 0 &= 2 \\
\mathrm{Ind}([1\ 1\ 1\ 0]) &= 2 + 1 &= 3
\end{aligned}$$

Basis lookup procedures (Lin tables)

- Lookup can therefore be done with two direct memory reads. This is a time and memory efficient approach (at least in many interesting cases).
- An alternative procedure is to build a hash list [const access time] or to perform a binary search [log access time].
- This becomes somewhat more involved when using spatial symmetries...

Symmetries

$$H = \sum_{i,j} J^{xy}_{i,j}(S^x_i S^x_j + S^y_i S^y_j) + J^z_{i,j} S^z_i S^z_j$$

Symmetries

- Consider a XXZ spin model on a lattice. What are the symmetries of the problem ?

$$H = \sum_{i,j} J^{xy}_{i,j} (S^x_i S^x_j + S^y_i S^y_j) + J^z_{i,j} S^z_i S^z_j$$

Symmetries

- Consider a XXZ spin model on a lattice. What are the symmetries of the problem ?

$$H = \sum_{i,j} J^{xy}_{i,j}(S^x_i S^x_j + S^y_i S^y_j) + J^z_{i,j} S^z_i S^z_j$$

- The Hamiltonian conserves total S^z, we can therefore work within a given S^z sector This easily implemented while constructing the basis, as we discussed before.

Symmetries

- Consider a XXZ spin model on a lattice. What are the symmetries of the problem ?

$$H = \sum_{i,j} J^{xy}_{i,j}(S^x_i S^x_j + S^y_i S^y_j) + J^z_{i,j} S^z_i S^z_j$$

- The Hamiltonian conserves total S^z, we can therefore work within a given S^z sector
  This easily implemented while constructing the basis, as we discussed before.

- The Hamiltonian is invariant under the space group, typically a few hundred elements.
  (in many cases = Translations x Pointgroup). Needs some technology to implement...

Symmetries

- Consider a XXZ spin model on a lattice. What are the symmetries of the problem ?

$$H = \sum_{i,j} J^{xy}_{i,j}(S^x_i S^x_j + S^y_i S^y_j) + J^z_{i,j} S^z_i S^z_j$$

- The Hamiltonian conserves total S^z, we can therefore work within a given S^z sector
This easily implemented while constructing the basis, as we discussed before.

- The Hamiltonian is invariant under the space group, typically a few hundred elements.
(in many cases = Translations x Pointgroup). Needs some technology to implement...

- At the Heisenberg point, the total spin is also conserved. It is however very difficult to combine the SU(2) symmetry with the lattice symmetries in a computationally useful way (non-sparse and computationally expensive matrices).

Symmetries

- Consider a XXZ spin model on a lattice. What are the symmetries of the problem ?

$$H = \sum_{i,j} J^{xy}_{i,j}(S^x_i S^x_j + S^y_i S^y_j) + J^z_{i,j} S^z_i S^z_j$$

- The Hamiltonian conserves total S^z, we can therefore work within a given S^z sector
This easily implemented while constructing the basis, as we discussed before.

- The Hamiltonian is invariant under the space group, typically a few hundred elements. (in many cases = Translations x Pointgroup). Needs some technology to implement...

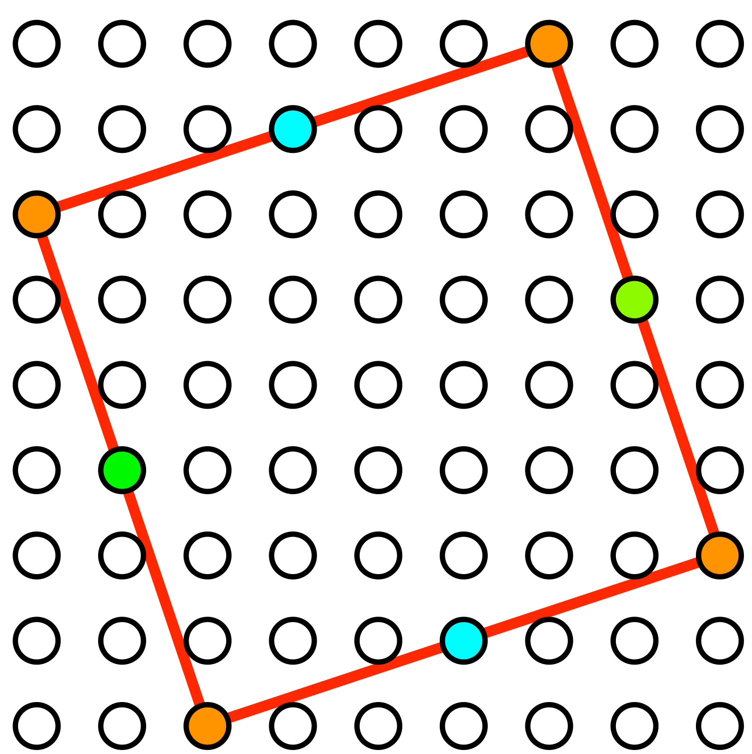
- At the Heisenberg point, the total spin is also conserved. It is however very difficult to combine the SU(2) symmetry with the lattice symmetries in a computationally useful way (non-sparse and computationally expensive matrices).

- At S^z=0 one can use the spin-flip (particle-hole) symmetry which distinguishes even and odd spin sectors at the Heisenberg point. Simple to implement.

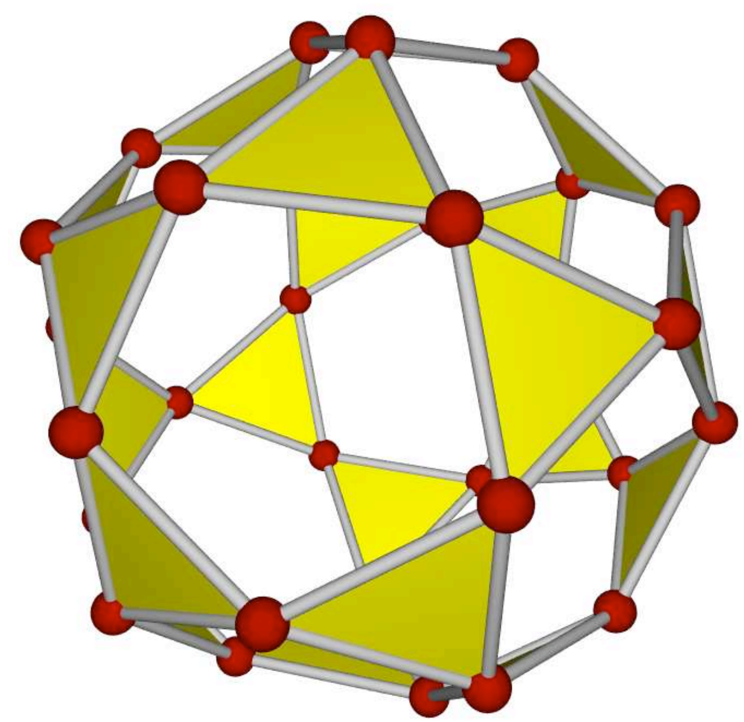
Spatial Symmetries

- Spatial symmetries are important for reduction of Hilbert space
- Symmetry resolved eigenstates teach a lot about the physics at work, dispersion of excitations, symmetry breaking tendencies, topological degeneracy, ...

40 sites square lattice
$T \otimes PG = 40 \times 4$ elements



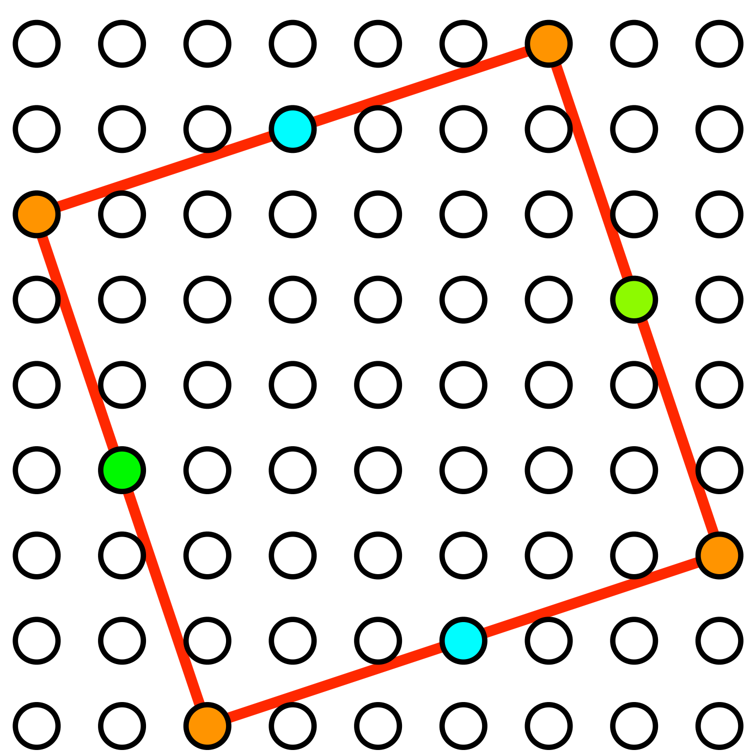
Icosidodecahedron (30 vertices)
I_h:120 elements



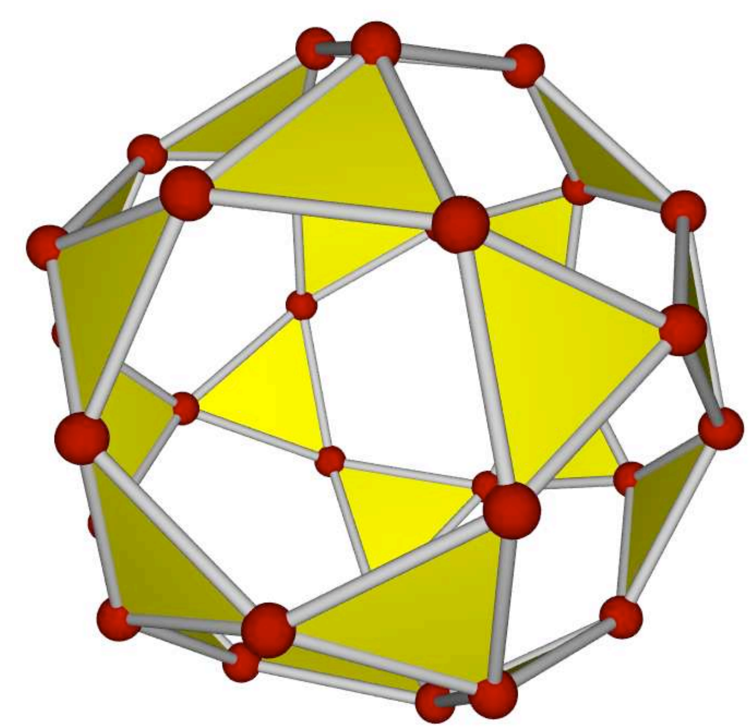
Spatial Symmetries

- Symmetries are sometimes not easily visible, use graph theoretical tools to determine symmetry group [nauty, grape].
- In an ED code a spatial symmetry operation is a site permutation operation.
(could become more complicated with spin-orbit interactions and multiorbital sites)

40 sites square lattice
$T \otimes PG$ =40 x 4 elements



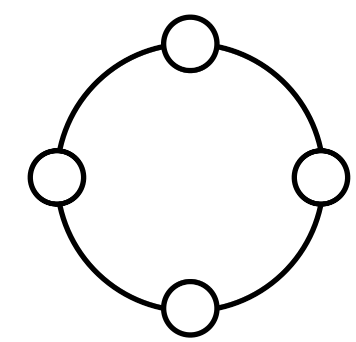
Icosidodecahedron (30 vertices)
I_h:120 elements



Spatial Symmetries: Building the basis

- Build a list of all allowed states satisfying the “diagonal” constraints, like particle number, total S^z, ...

- for each state we apply all symmetry operations and keep the state as a representative if it has the smallest integer representation among all generated states in the orbit.
  Example: 4 site ring with cyclic translation T, S^z=3/2 sector



$$\begin{aligned} T^0([0\ 1\ 1\ 1]) &\rightarrow [0\ 1\ 1\ 1] \\ T^1([0\ 1\ 1\ 1]) &\rightarrow [1\ 0\ 1\ 1] \\ T^2([0\ 1\ 1\ 1]) &\rightarrow [1\ 1\ 0\ 1] \\ T^3([0\ 1\ 1\ 1]) &\rightarrow [1\ 1\ 1\ 0] \end{aligned}$$

keep state

$$\begin{aligned} T^0([1\ 0\ 1\ 1]) &\rightarrow [1\ 0\ 1\ 1] \\ T^1([1\ 0\ 1\ 1]) &\rightarrow [1\ 1\ 0\ 1] \\ T^2([1\ 0\ 1\ 1]) &\rightarrow [1\ 1\ 1\ 0] \\ T^3([1\ 0\ 1\ 1]) &\rightarrow [0\ 1\ 1\ 1] \end{aligned}$$

discard state

...

Spatial Symmetries: Building the basis

$$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g) |g(r)\rangle$$

$$\mathcal{N} = \sqrt{\sum_{g \in G, g(r) = r} \chi(r)}$$

Spatial Symmetries: Building the basis

- For one-dimensional representations χ of the spatial symmetry group:
  - “Bloch” state $$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g) |g(r)\rangle$$
  - Norm of the state is given as: $$\mathcal{N} = \sqrt{\sum_{g \in G, g(r) = r} \chi(r)}$$
  - The norm (and therefore the state itself) can vanish if it has a nontrivial stabilizer combined with a nontrivial representation χ.

Spatial Symmetries: Building the basis

- For one-dimensional representations χ of the spatial symmetry group:
  - "Bloch" state
    $$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g) |g(r)\rangle$$
  - Norm of the state is given as:
    $$\mathcal{N} = \sqrt{\sum_{g \in G, g(r)=r} \chi(r)}$$
  - The norm (and therefore the state itself) can vanish if it has a nontrivial stabilizer combined with a nontrivial representation χ.
- Example: 4 site S=1/2 ring with cyclic translations:

Spatial Symmetries: Building the basis

- For one-dimensional representations χ of the spatial symmetry group:
  - "Bloch" state $$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g) |g(r)\rangle$$
  - Norm of the state is given as: $$\mathcal{N} = \sqrt{\sum_{g \in G, g(r)=r} \chi(r)}$$
  - The norm (and therefore the state itself) can vanish if it has a nontrivial stabilizer combined with a nontrivial representation χ.
- Example: 4 site S=1/2 ring with cyclic translations:

	$K = 0$
$S^z=2$	$\|1\ 1\ 1\ 1\rangle, \mathcal{N} = 2$
$S^z=1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N} = 1$
$S^z=0$	$\|0\ 1\ 0\ 1\rangle, \mathcal{N} = \sqrt{2}$
	$\|0\ 0\ 1\ 1\rangle, \mathcal{N} = 1$

Spatial Symmetries: Building the basis

- For one-dimensional representations χ of the spatial symmetry group:
  - “Bloch” state $$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g)|g(r)\rangle$$
  - Norm of the state is given as: $$\mathcal{N} = \sqrt{\sum_{g \in G, g(r)=r} \chi(r)}$$
  - The norm (and therefore the state itself) can vanish if it has a nontrivial stabilizer combined with a nontrivial representation χ.
- Example: 4 site S=1/2 ring with cyclic translations:

	$K = 0$	$K = \pm\pi/2$
$S^z=2$	$\|1\ 1\ 1\ 1\rangle, \mathcal{N} = 2$	
$S^z=1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N} = 1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N} = 1$
$S^z=0$	$\|0\ 1\ 0\ 1\rangle, \mathcal{N} = \sqrt{2}$	
	$\|0\ 0\ 1\ 1\rangle, \mathcal{N} = 1$	$\|0\ 0\ 1\ 1\rangle, \mathcal{N} = 1$

Spatial Symmetries: Building the basis

- For one-dimensional representations χ of the spatial symmetry group:
  - "Bloch" state $$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g) |g(r)\rangle$$
  - Norm of the state is given as: $$\mathcal{N} = \sqrt{\sum_{g \in G, g(r)=r} \chi(r)}$$
  - The norm (and therefore the state itself) can vanish if it has a nontrivial stabilizer combined with a nontrivial representation χ.
- Example: 4 site S=1/2 ring with cyclic translations:

	$K=0$	$K=\pm\pi/2$	$K=\pi$
S^z=2	$\|1\ 1\ 1\ 1\rangle, \mathcal{N}=2$		
S^z=1	$\|0\ 1\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N}=1$
S^z=0	$\|0\ 1\ 0\ 1\rangle, \mathcal{N}=\sqrt{2}$		$\|0\ 1\ 0\ 1\rangle, \mathcal{N}=\sqrt{2}$
	$\|0\ 0\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 0\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 0\ 1\ 1\rangle, \mathcal{N}=1$

Spatial Symmetries: Building the basis

- For one-dimensional representations χ of the spatial symmetry group:
  - "Bloch" state $$|\tilde{r}\rangle = \frac{1}{\mathcal{N}\sqrt{|G|}} \sum_{g \in G} \chi(g) |g(r)\rangle$$
  - Norm of the state is given as: $$\mathcal{N} = \sqrt{\sum_{g \in G, g(r)=r} \chi(r)}$$
  - The norm (and therefore the state itself) can vanish if it has a nontrivial stabilizer combined with a nontrivial representation χ.
- Example: 4 site S=1/2 ring with cyclic translations:

	$K=0$	$K=\pm\pi/2$	$K=\pi$	**2^4=16**
S^z=2	$\|1\ 1\ 1\ 1\rangle, \mathcal{N}=2$			1+1
S^z=1	$\|0\ 1\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 1\ 1\ 1\rangle, \mathcal{N}=1$	4+4
S^z=0	$\|0\ 1\ 0\ 1\rangle, \mathcal{N}=\sqrt{2}$		$\|0\ 1\ 0\ 1\rangle, \mathcal{N}=\sqrt{2}$	2
	$\|0\ 0\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 0\ 1\ 1\rangle, \mathcal{N}=1$	$\|0\ 0\ 1\ 1\rangle, \mathcal{N}=1$	4

The Hamiltonian Matrix

The Hamiltonian Matrix

- Now that we have a list of representatives and their norms, can we calculate the matrix elements of the Hamiltonian ? $\langle \tilde{s}|H|\tilde{r}\rangle = ?$

- Let us look at an elementary, non-branching term in the Hamiltonian:

$$h^\alpha|r\rangle = h^\alpha(r)|s\rangle$$

- We can now calculate the matrix element $\langle \tilde{s}|h^\alpha|\tilde{r}\rangle$ without double expanding the Bloch states:

$$\langle \tilde{s}|h^\alpha|\tilde{r}\rangle = \frac{\mathcal{N}_s}{\mathcal{N}_r}\chi(g^*)h^\alpha(r)$$

- key algorithmic problem: given a possibly non-representative $|s\rangle$, how do we find the associated representative $|\tilde{s}\rangle$, as well as a symmetry element g^* relating $|s\rangle$ to $|\tilde{s}\rangle$?

The Hamiltonian Matrix

- key algorithmic problem: given a possibly non-representative $|s\rangle$, how do we find the associated representative $|\tilde{s}\rangle$, as well as a symmetry element g^* relating $|s\rangle$ to $|\tilde{s}\rangle$?
  - Brute force: loop over all symmetry operations applied on $|s\rangle$ and retain $|\tilde{s}\rangle$ and g^*. This is however often not efficient (many hundred symmetries).
  - Prepare a lookup list, relating each allowed configuration with the index of its representative, and also the associated group element linking the two. Gives fast implementation, but needs a list of the size of the non spatially-symmetrized Hilbert space.
  - For specific lattices and models (Hubbard models) clever tricks exist which factorize the symmetry group into a sublattice conserving subgroup times a sublattice exchange. They give $|\tilde{s}\rangle$ fast, then a hash or binary search is needed to locate $|\tilde{s}\rangle$ in the list of representatives in order to get its index.

Hamiltonian Matrix Storage

- Different possibilities exist:
  - Store hamiltonian matrix elements in memory in a sparse matrix format
    Fast matrix vector multiplies, but obviously limited by available memory.
  - Store hamiltonian matrix elements on disk in a sparse matrix format.
    In principle possible due to the vast disk space available, but I/O speed is much slower than main memory access times. Difficult to parallelize.
  - Recalculate the hamiltonian matrix elements in each iterations “on the fly”.
    Needed for the cutting edge simulations, where the whole memory is used by the Lanczos vectors. Can be parallelized on most architectures.

The Linear Algebra Backend

Linear Algebra:
The most popular: Lanczos Algorithm

- Lanczos Algorithm (C. Lanczos, 1950)

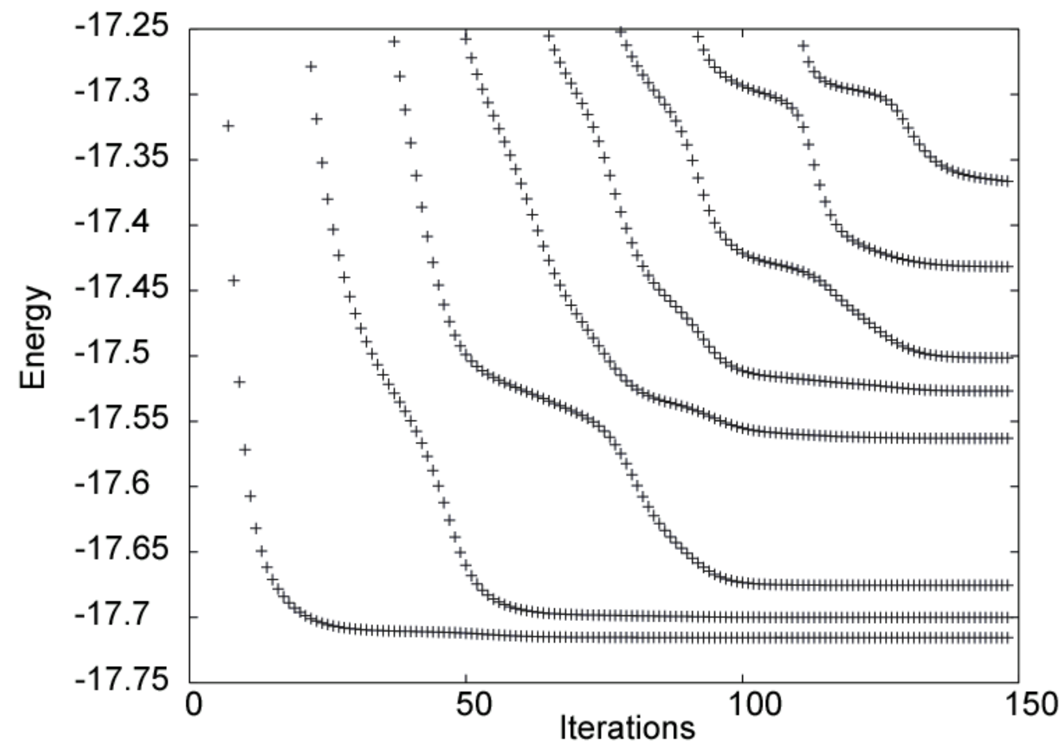
Three vector recursion

$$
\begin{aligned}
|\phi'\rangle &= \boxed{H|\phi_n\rangle} - \beta_n|\phi_{n-1}\rangle\,,\\
\alpha_n &= \langle\phi_n|\phi'\rangle\,,\\
|\phi''\rangle &= |\phi'\rangle - \alpha_n|\phi_n\rangle\,,\\
\beta_{n+1} &= ||\phi''|| = \sqrt{\langle\phi''|\phi''\rangle}\,,\\
|\phi_{n+1}\rangle &= |\phi''\rangle/\beta_{n+1}\,,
\end{aligned}
$$

$$
\tilde{H}_N = \begin{bmatrix}
\alpha_0 & \beta_1 & 0 & \ldots\ldots\ldots\ldots & & 0\\
\beta_1 & \alpha_1 & \beta_2 & 0 & \ldots\ldots & 0\\
0 & \beta_2 & \alpha_2 & \beta_3 & 0 & 0\\
 & & \ddots & \ddots & \ddots & \\
0 & \ldots & 0 & \beta_{N-2} & \alpha_{N-2} & \beta_{N-1}\\
0 & \ldots\ldots & & 0 & \beta_{N-1} & \alpha_{N-1}
\end{bmatrix}
$$

- Eigenvalues of H_N converge rapidly towards eigenvalues of H.

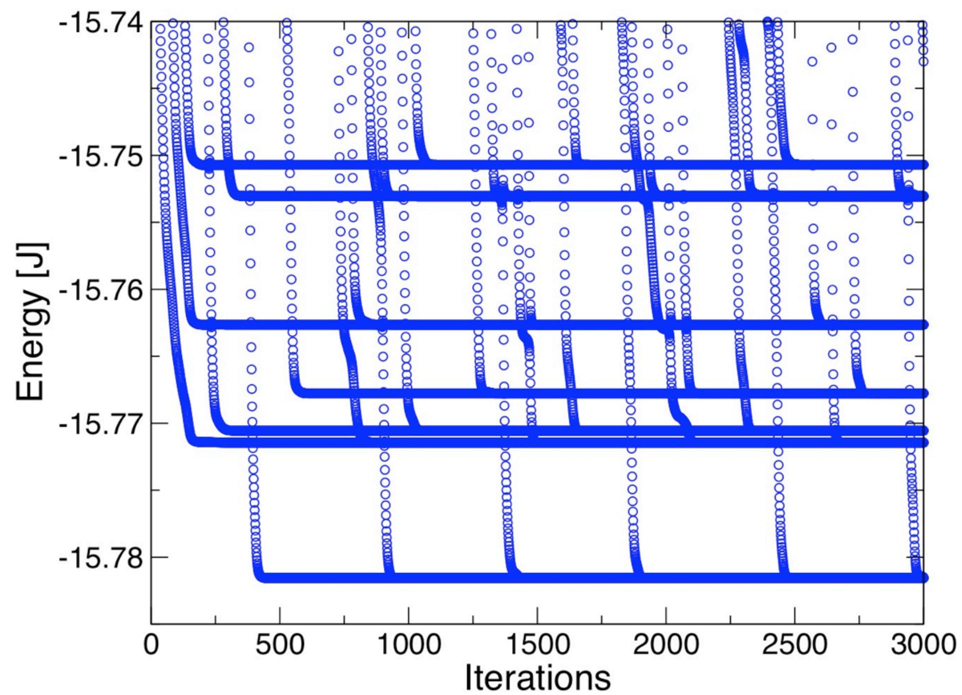
- Once desired eigenvalue is converged, restart recursion and assemble the eigenvector.



very quick convergence for extremal eigenvalues !

Linear Algebra:
Lanczos Algorithm

- Once the ground state has converged, the vectors in the recursion tend to lose their orthogonality. As a consequence fake new eigenvalues show up in the approximate spectrum. These can be removed by heuristic techniques

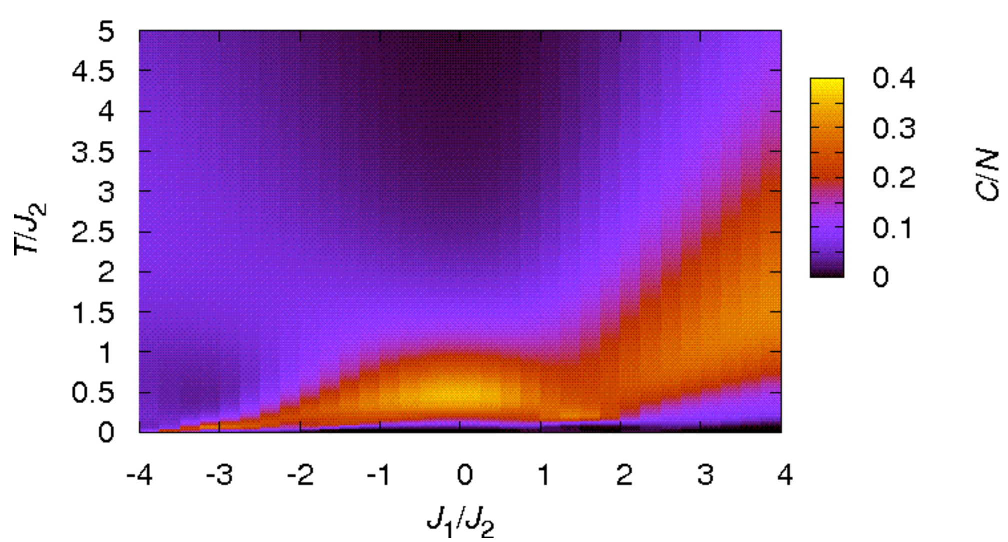


- Degeneracies of eigenvalues can not be resolved by construction. For this task one would need a band lanczos or the (Jacobi-)Davidson technique. However multiply degenerate eigenvalues are converged.

- Checkpointing is useful when performing large-scale simulations.

Full Diagonalization: Thermodynamics

- Lapack / Householder complete diagonalization of the spectrum.
- Calculate partition function and all the thermodynamic quantities you want, often the only pedestrian method available for frustrated systems.
- Symmetries are also very important, because the computational requirements scale as $O(D^3)$, where D is the dimension of the block Hilbert space. Typical D's for a workstation are a few 1'000, up to a few 100'000 on supercomputers.



F. Heidrich-Meisner, A. Honecker, T. Vekua, Phys. Rev. B **74**, 020403(R) (2006).

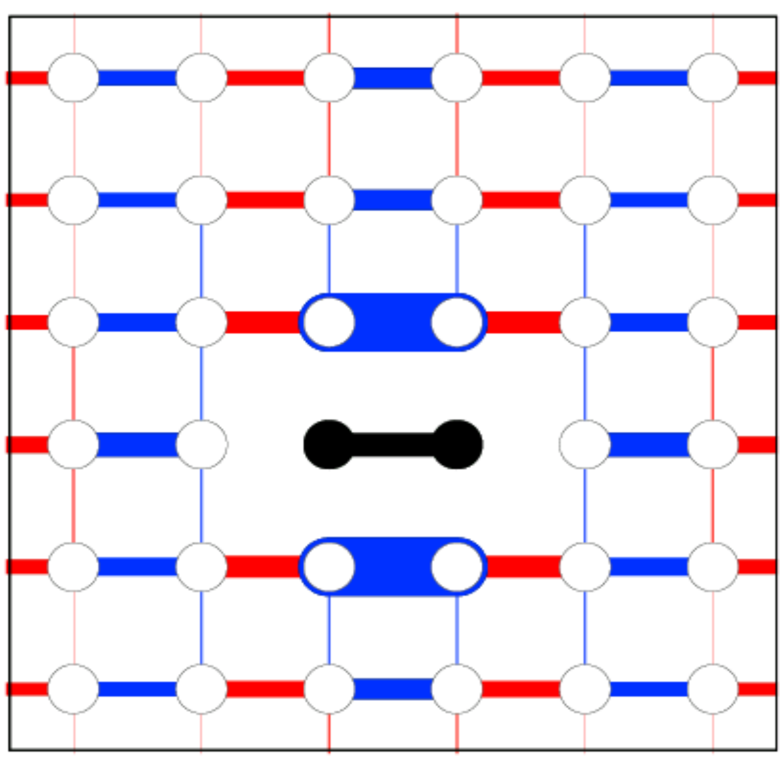
Observables

Observables

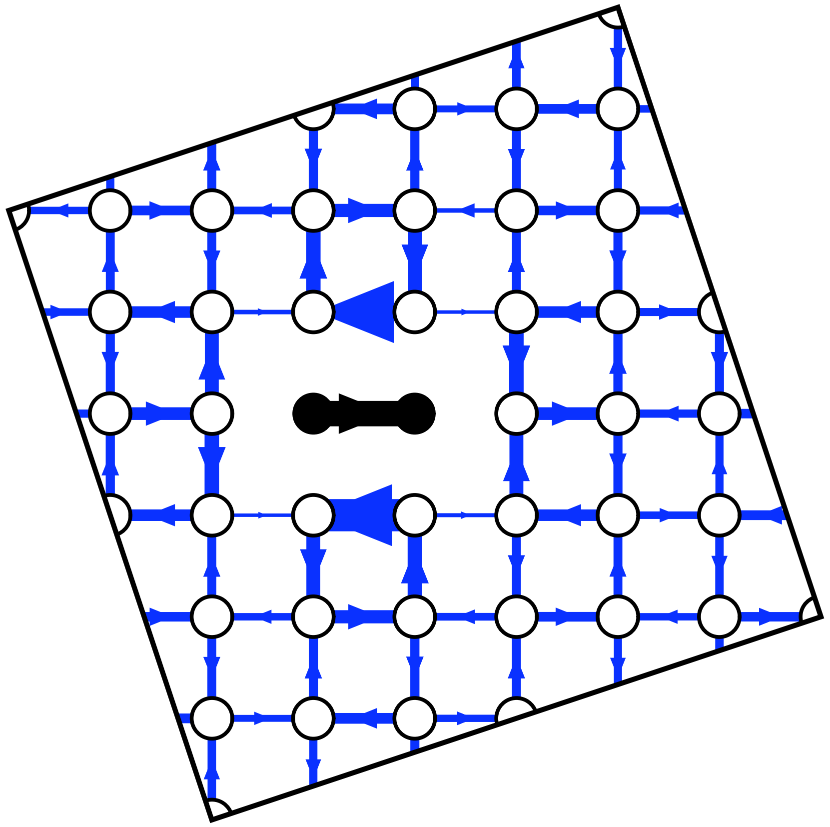
- In principle once can calculate any correlation function, since one has access to the full many body wave functions. When using spatial symmetries, the correlation functions need to be properly symmetrized too.

- Complicated correlation functions occur in frustrated systems:

Dimer-dimer correlations

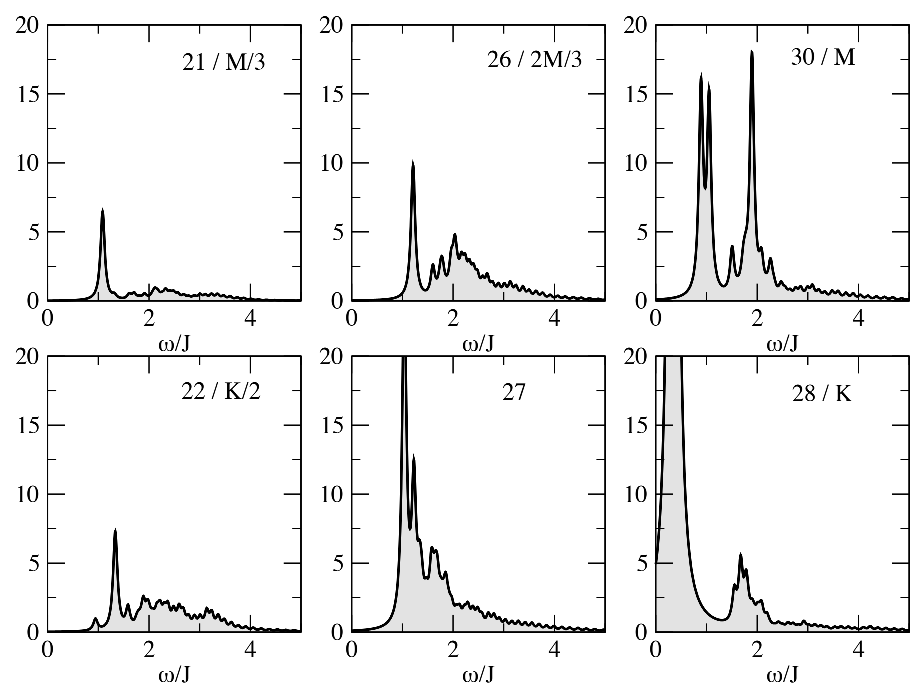


Spin current correlations



Frequency Dynamics

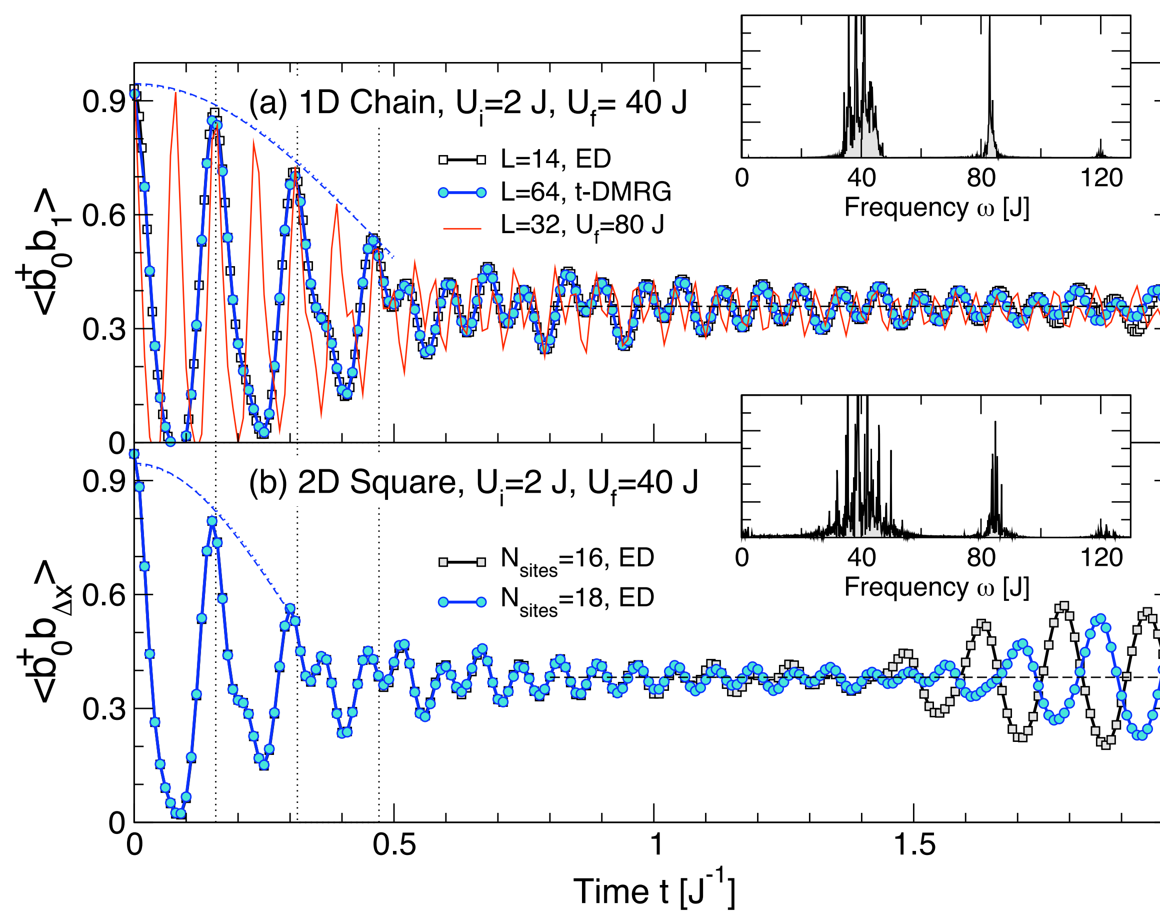
- $$G_A(\omega + i\eta) = \langle\psi|A^\dagger \frac{1}{E_0 + \omega + i\eta - H} A|\psi\rangle \quad A = S^\alpha(\mathbf{q}), c_{\mathbf{k}}, \ldots$$
- Generate Krylov space of $A|\psi\rangle$
  Use continued fraction used to invert $(E_0 + \omega + i\eta - H)$
- Triangular Lattice Spin Dynamics in zero field



AML unpublished

Exact Diagonalization
Real-Time Dynamics

- It is expensive to obtain the full propagator $\exp[-itH]$
- Krylov methods exist to approximate the propagator for a given state $|\psi(0)\rangle$
  One can get the time propagated state $|\psi(t)\rangle$ with only $|v\rangle = H|u\rangle$ operations.
- Example: time evolution of a strongly correlated quantum systems after an abrupt change in the parameters in the Hamiltonian. Revivals and Relaxation.

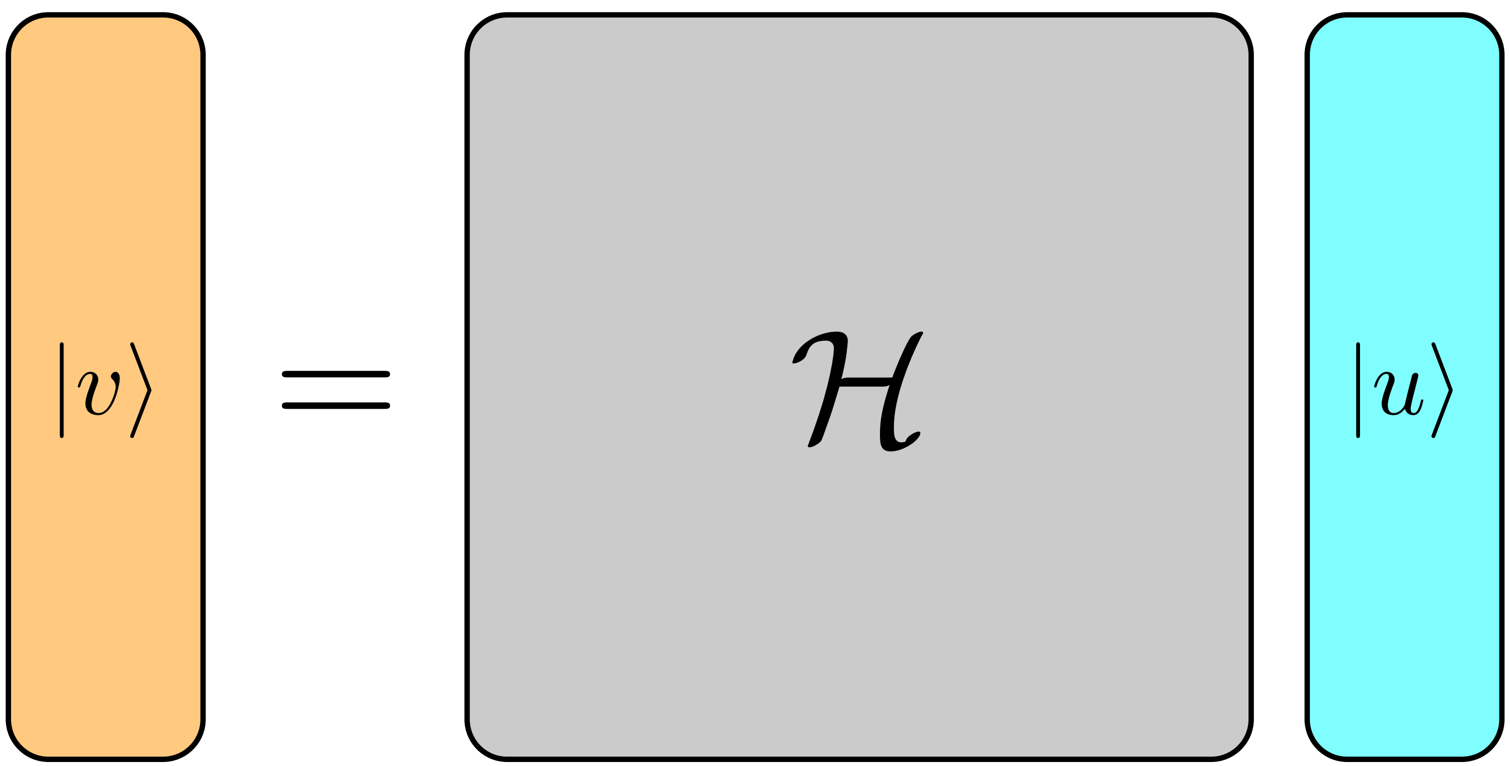


C. Kollath, AML, E. Altman, PRL 2007

Parallelization Strategies

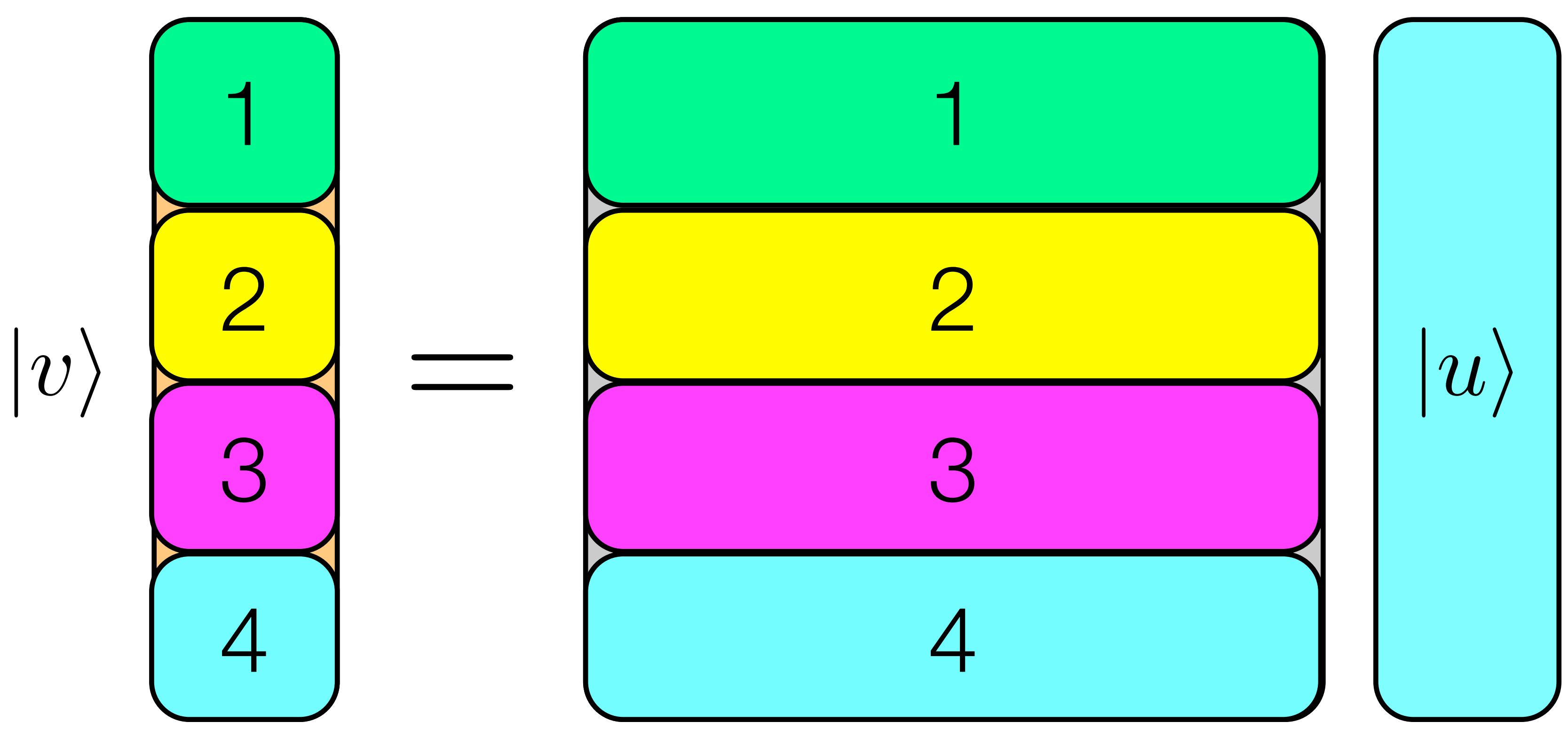
Parallelization:
Shared memory nodes

- In the Lanczos algorithm the heaviest part is the elementary matrix-vector multiplication.
- In a matrix-free formulation this part can easily be parallelized using OpenMP pragmas in the code, even on your multi-core workstation.
  Choose the right strategy between pull and push !



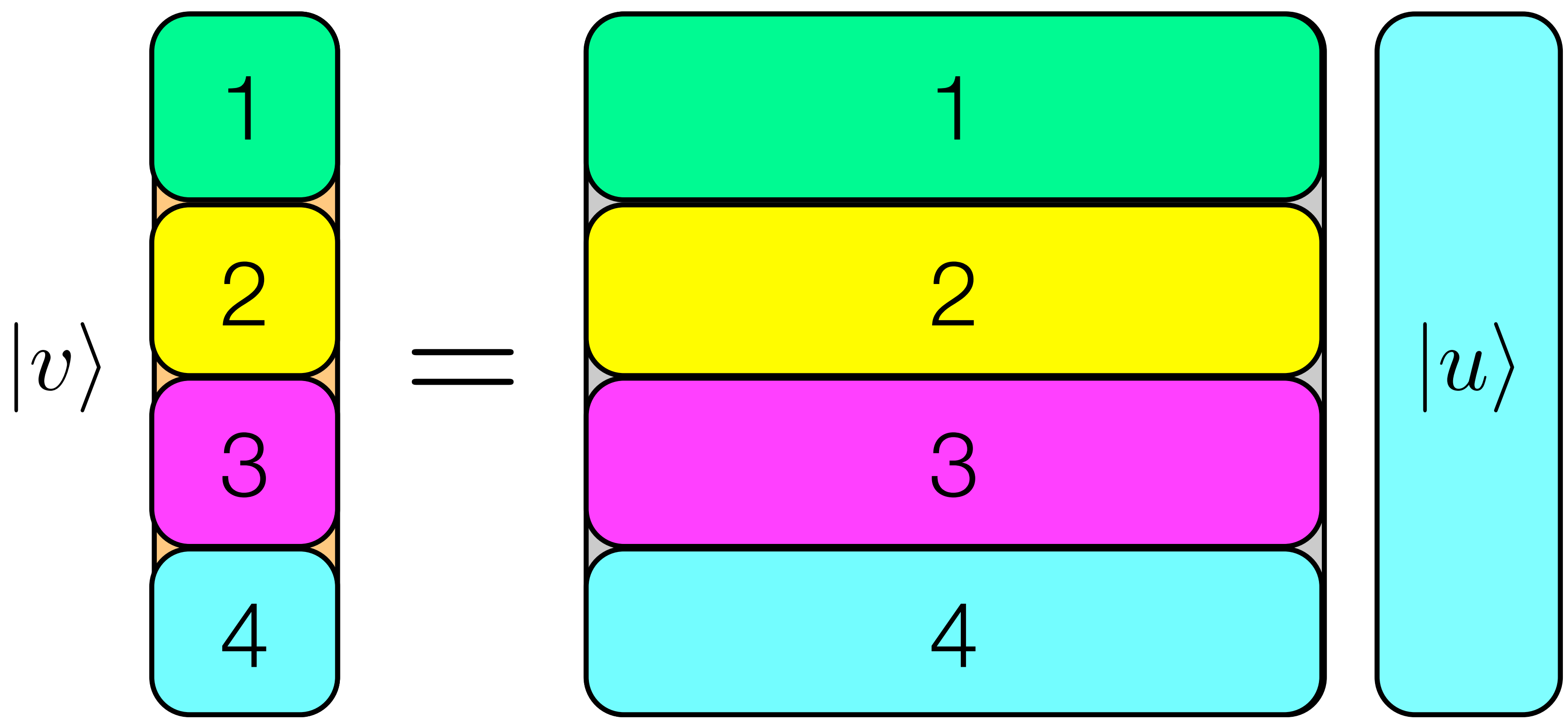
Parallelization:
Shared memory nodes

- In the Lanczos algorithm the heaviest part is the elementary matrix-vector multiplication.
- In a matrix-free formulation this part can easily be parallelized using OpenMP pragmas in the code, even on your multi-core workstation.
  Choose the right strategy between pull and push !



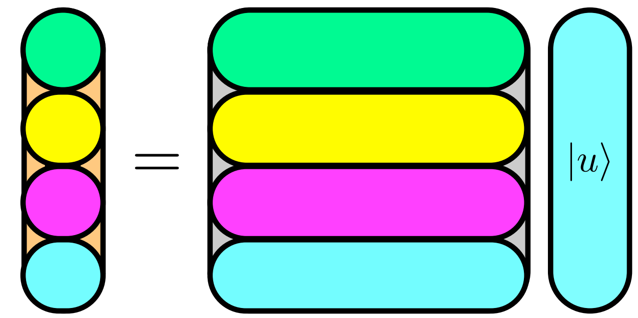
Parallelization:
Shared memory nodes

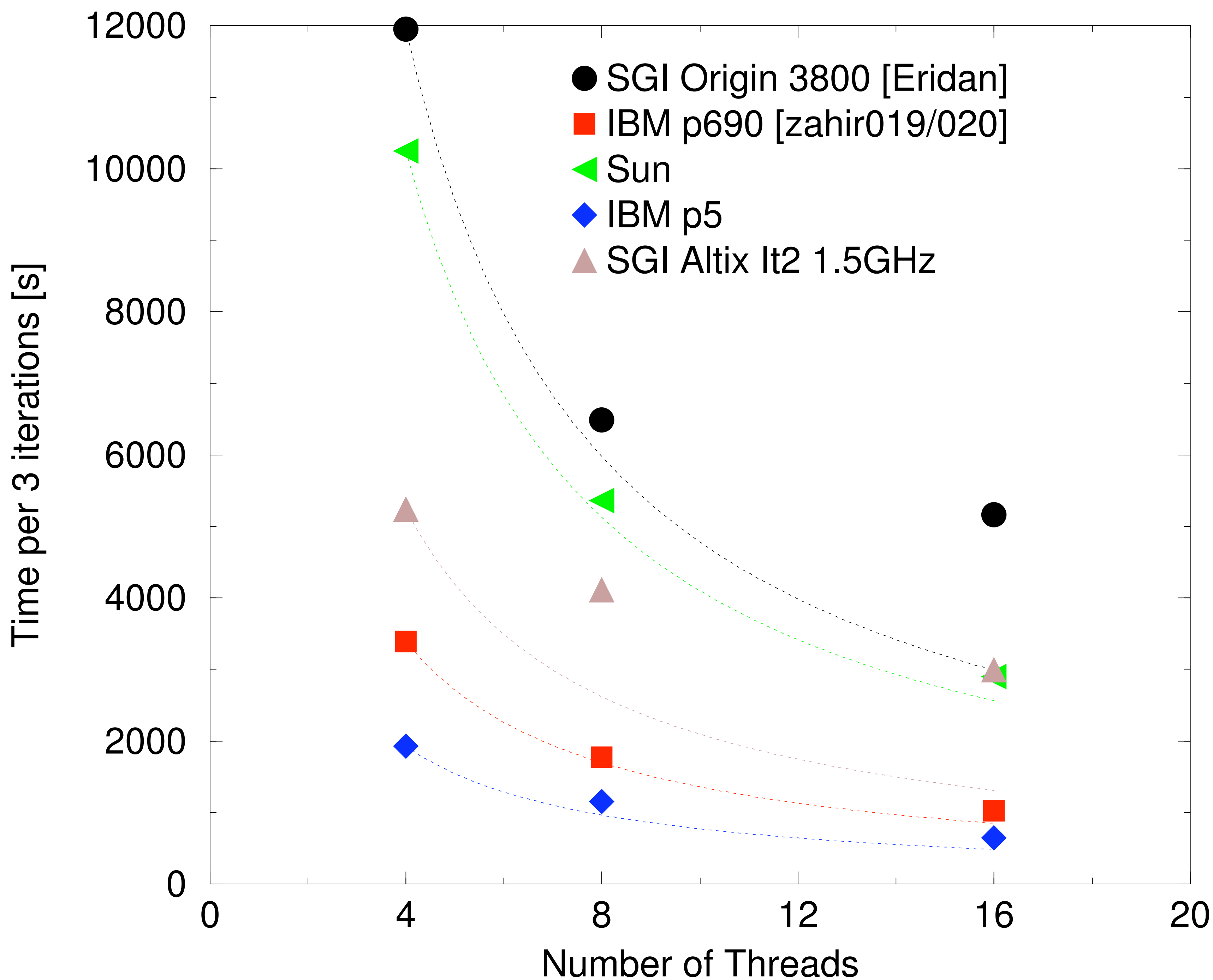
- In the Lanczos algorithm the heaviest part is the elementary matrix-vector multiplication.
- In a matrix-free formulation this part can easily be parallelized using OpenMP pragmas in the code, even on your multi-core workstation.
  Choose the right strategy between pull and push !



In this parallelization we have uncritical concurrent reads, but no concurrent updates of vector v.

Parallelization: Shared memory nodes





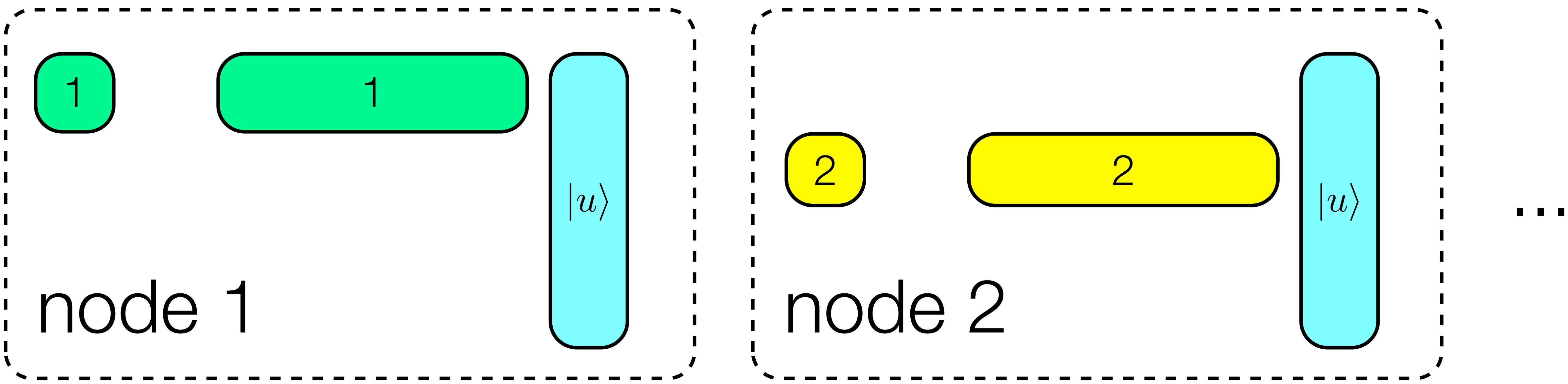
- scales well up to a few ten threads on “memory uniform” SMP machines.

Parallelization: Distributed memory nodes

- For some classes of problems the Hilbert space size is not too big, but the vast number of matrix elements is a challenge. [ED in momentum space formulation & Quantum Hall problems]
- These problems can be OpenMP parallelized, but are also suitable for large scale Message passing parallelization.

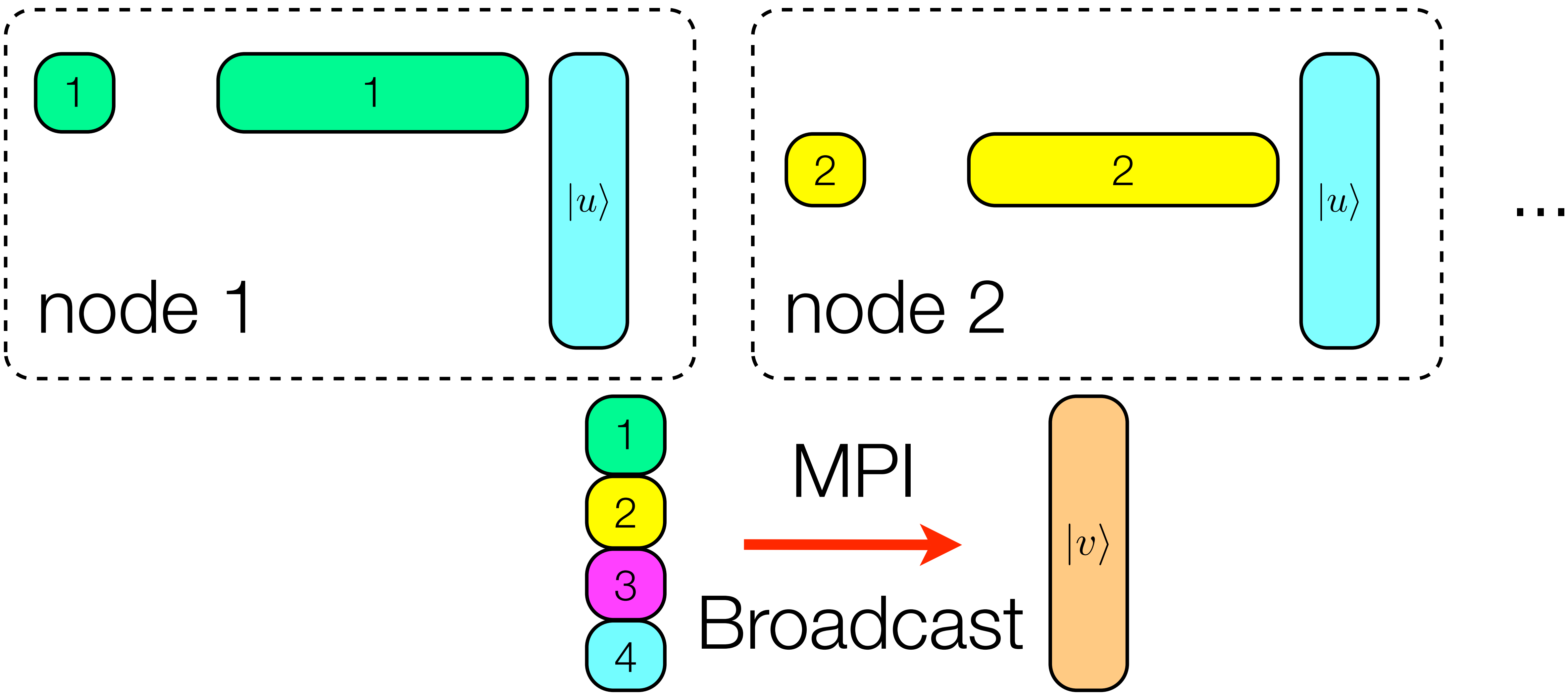
Parallelization:
Distributed memory nodes

- For some classes of problems the Hilbert space size is not too big, but the vast number of matrix elements is a challenge. [ED in momentum space formulation & Quantum Hall problems]
- These problems can be OpenMP parallelized, but are also suitable for large scale Message passing parallelization.



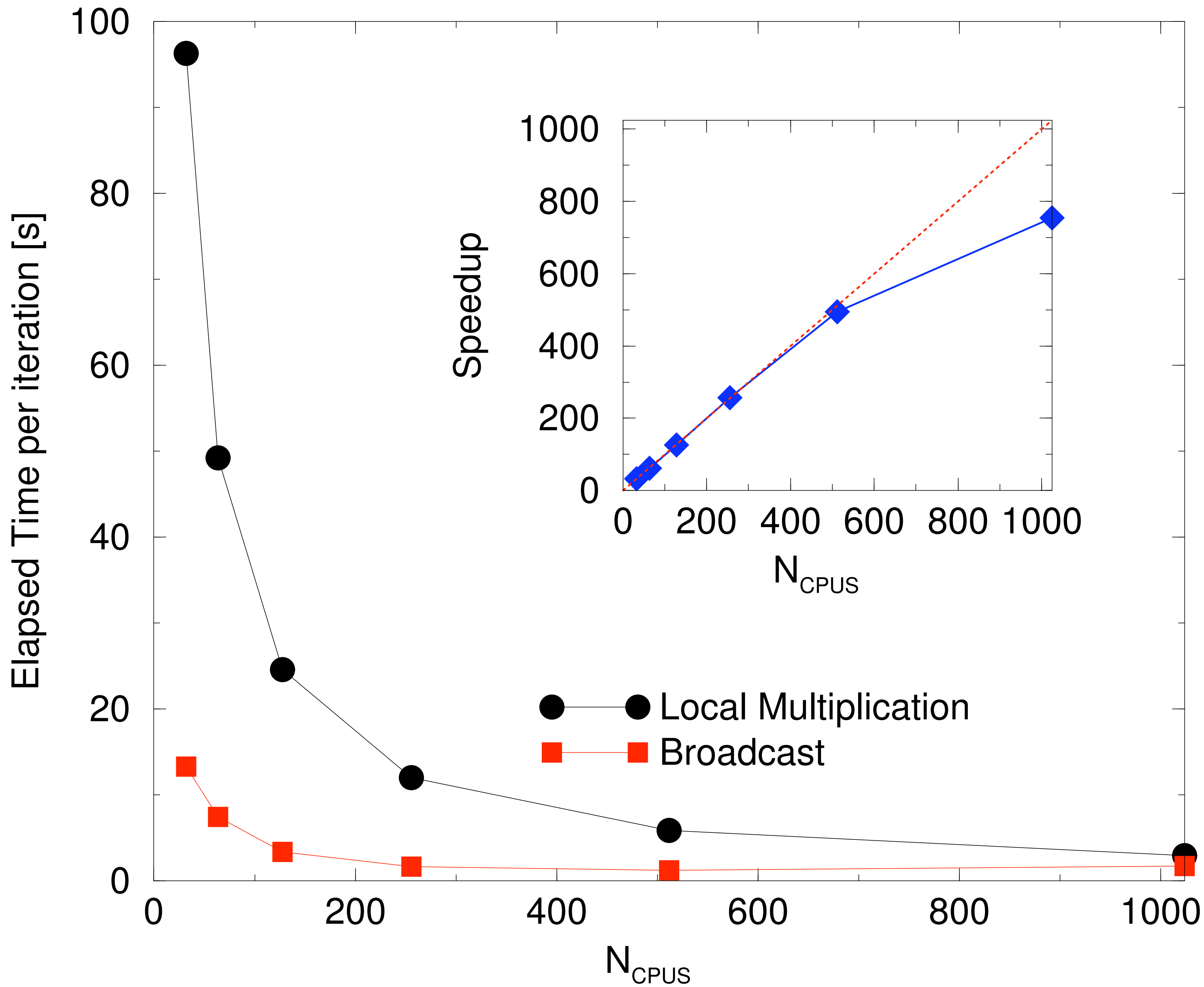
Parallelization:
Distributed memory nodes

- For some classes of problems the Hilbert space size is not too big, but the vast number of matrix elements is a challenge. [ED in momentum space formulation & Quantum Hall problems]
- These problems can be OpenMP parallelized, but are also suitable for large scale Message passing parallelization.



Parallelization:
Distributed memory nodes

- Strong scaling example RG-ED: matrix dimension 10 million performed on a 1024 node Cray XT-3 machine: speedup of ≈ 800 on 1024 procs

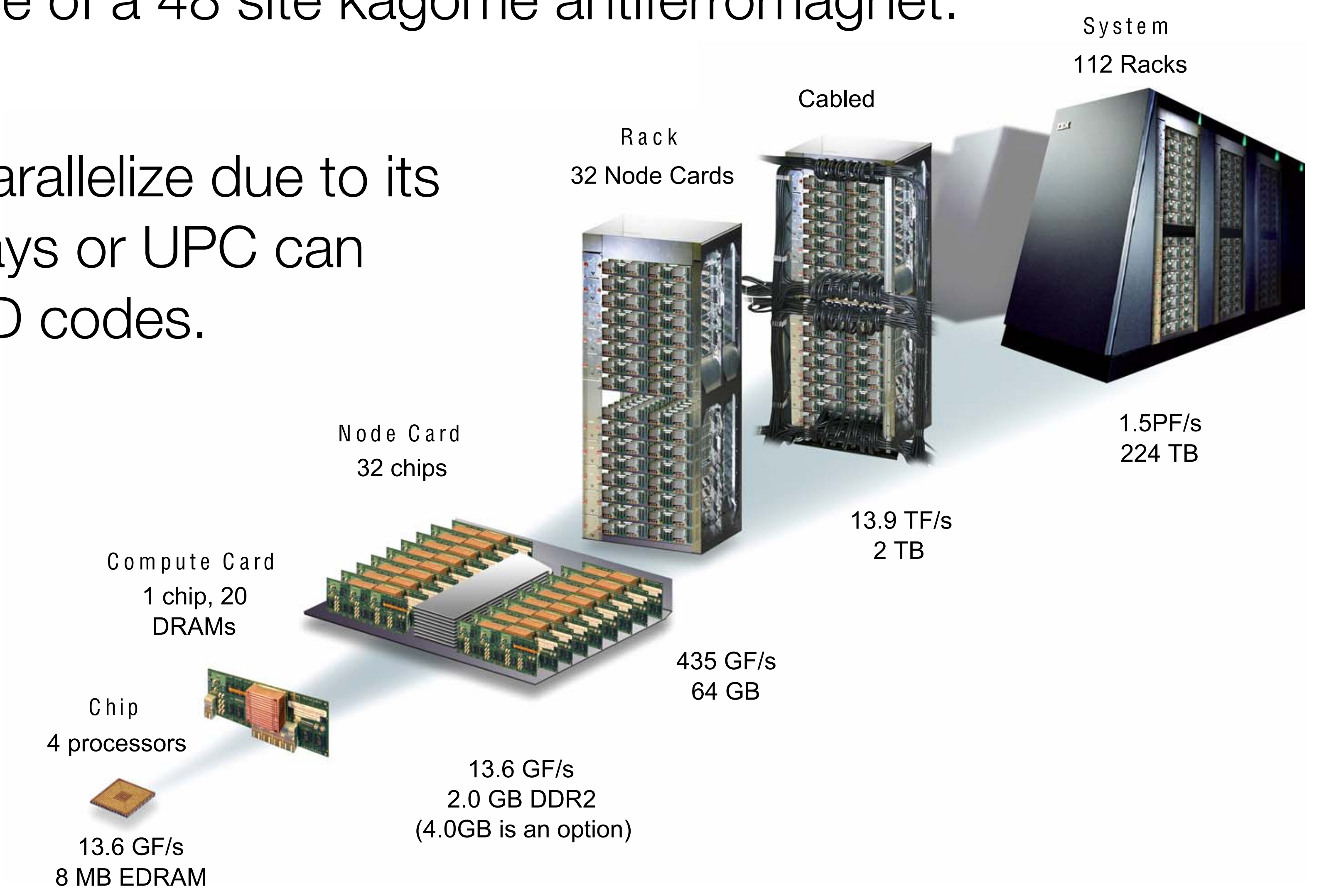


Parallelization:
How to harness the petaflop computers ?

- Cutting edge petaflop systems have a huge number of core, but only a moderate amount of node-local memory.

- Next generation ED codes need to be developed in order to attack e.g. the 80 billion Hilbert space of a 48 site kagome antiferromagnet.

- Problem remains difficult to parallelize due to its all-to-all structure. Global Arrays or UPC can help developing distributed ED codes.



Exact Diagonalization Literature

- N. Laflorencie & D. Poilblanc,

  “Simulations of pure and doped low-dimensional spin-1/2 gapped systems”

  Lecture Notes in Physics **645**, 227 (2004).

- R.M. Noack & S. Manmana,

  “*Diagonalization- and Numerical Renormalization-Group-Based Methods for Interacting Quantum Systems*”,

  AIP Conf. Proc. **789**, 93 (2005).

- A. Weisse, H. Fehske

  “Exact Diagonalization Techniques”

  Lecture Notes in Physics 739, 529 (2008).

- A. Läuchli

  ”Numerical Simulations of Frustrated Systems”

  to appear in “Highly Frustratred Magnetism”, Eds. Lacroix Mendels, Mila, (2009).

Thank you !