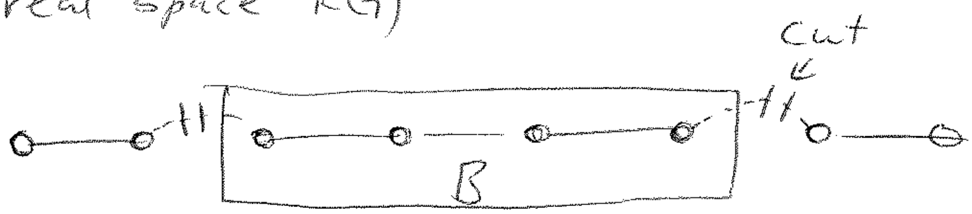
Wilson Numerical RG

Goal of method: Find ground and low-lying excited states.

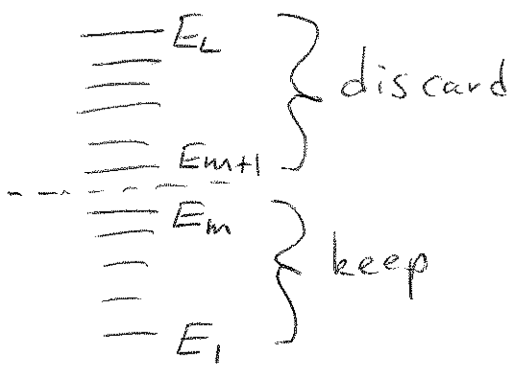
First applied to Kondo impurity problem (K.G. Wilson, Rev. Mod. Phys. V47, 773 (1975)). In that context it is complicated by various changes of basis to map the system onto a 1D half-lattice

We consider here the numerical method applied to a full 1D lattice:
(real space RG)



Procedure

1. Diagonalize block B.
2. Form partial matrix of eigenvectors O, containing m lowest energy states v_i.



$$O = \begin{array}{c} \\ N \end{array} \overset{\leftarrow m \rightarrow}{\begin{pmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \cdots & v_m \\ \vdots & \vdots & & \vdots \end{pmatrix}}$$

Leave out
$v_{m+1} \cdots v_L$
v_1 = ground state
v_2 = 1st excited state

3. Change basis and truncate all operators describing B, getting new block B'.

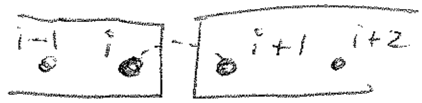
$$\underset{m\times m}{H'} = \underset{m\times N}{O^T} \overset{N\times N}{H} \underset{N\times m}{O}; \qquad S_i^{z\prime} = O^T S_i^z O, \text{ etc.}$$

(Leave original site basis behind.) In some sense $B' \approx B$.

4. Combine two adjacent B''s.

$$\underset{m^2}{B''} = \underset{m}{B'} \otimes \underset{m}{B'}; \qquad |i_1 i_2\rangle = |i_1\rangle|i_2\rangle$$

$$H'' = H' \otimes \mathbf{1} + \mathbf{1} \otimes H' + \text{connecting terms}$$

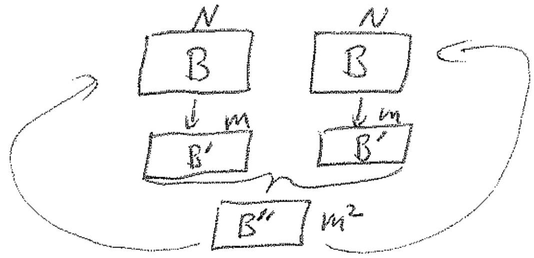
Typical connecting term: $S_i^z S_{i+1}^z \rightarrow$ 

$$[S_i^z S_{i+1}^z]'' = S_i^{z\prime} \otimes S_{i+1}^{z}{}'$$

translation

$$[S_i^z S_{i+1}^z]''_{j_1 j_2, j_1' j_2'} = [S_i^{z\prime}]_{j_1 j_1'} [S_{i+1}^{z}{}']_{j_2 j_2'}$$

5. Replace B by B'' and iterate.



After 1st iter, $N = m^2$

What justifies the truncation?

1. We want the ground state and we are throwing out high energy states (of small blocks).

2. In limit "connecting terms" are small, perturbation theory justifies it.

3. Detailed analysis of structure of H for impurity problems.

Where has it been used?

Successes

- Impurity problems (Kondo, Anderson impurity, two Kondo impurities).

Failures

- All lattice models

Test case–1D particle in a box

Continuum version: $H = -\frac{\partial^2}{\partial x^2}$; $\psi(0) = \psi(L) = 0$.

Lattice version:

Ground state: 

excited state: 

$$H = \begin{pmatrix} 2 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ 0 & & & & \ddots & \end{pmatrix}$$

Exercise:
find exact E-vals.

This problem was studied as a test case for why RG fails by Wilson in 1986 (unpublished).

In this 1 particle problem, instead of adding blocks using direct products $\otimes$, we use direct sums $\oplus$. Number of states $= L$, not 2^L or 4^L.

Before $\psi_{ij} = u_i v_j$
now $\psi = \begin{pmatrix} u \\ \hline v \end{pmatrix}$

<u>Procedure</u>

$$H_{\text{system}} = \begin{pmatrix} H & T & & & & 0 \\ T^\dagger & H & T & & & \\ & T^\dagger & H & T & & \\ & & T^\dagger & H & T & \\ 0 & & & & \ddots & \end{pmatrix}$$

Initially $H = (2)$ and $T = (-1)$.

1. Combine two blocks:

$$H' = \begin{pmatrix} H & T \\ T^\dagger & H \end{pmatrix} \qquad T' = \begin{pmatrix} 0 & 0 \\ T & 0 \end{pmatrix}$$

2. Diagonalize H', getting eigenvectors V_ℓ

3. Form matrix O

$$O = \begin{pmatrix} \vdots & \vdots & & \vdots \\ V_1 & V_2 & \dots & V_m \\ \vdots & \vdots & & \vdots \end{pmatrix}$$

discard
V_{m+1} to V_N
(After 1st couple of iters, $N = 2m$.)

4. Change basis and truncate:

O assumed real

$$\underset{m\times m}{H''} = \underset{m\times N}{O^T} \underset{N\times N}{H} \underset{N\times m}{O} \qquad T'' = O^T T O$$

Really we have

$$O_{b.g} = \begin{pmatrix} O & & & \emptyset \\ & O & & \\ & & O & \\ \emptyset & & & O \end{pmatrix}$$

slope $\neq -1$, not on diag.

5. Replace H and T by H'' and T'' and iterate.

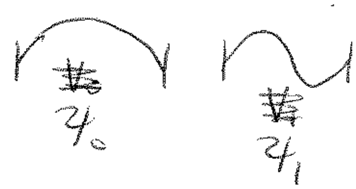
How does it do?

Test calculation: 10 blockings, keeping $m = 8$ states:

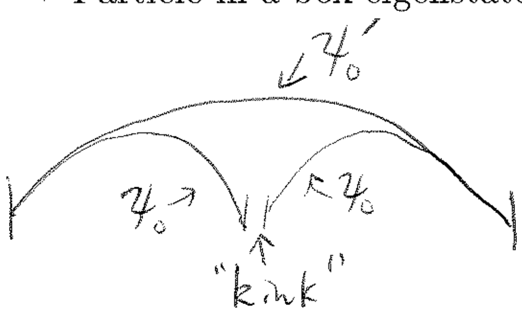
	Exact	RG
E_0	2.351×10^{-6}	1.9207×10^{-2}
E_1	9.403×10^{-6}	1.9209×10^{-2}
E_2	2.116×10^{-5}	1.9214×10^{-2}
E_3	3.761×10^{-5}	1.9217×10^{-2}

It performs terribly. Why? Look at continuum states.

Isolating a block sets ψ to 0 at the edges (fixed BCs).



→ Particle-in-a-box eigenstates.



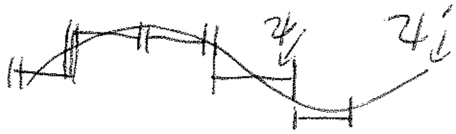
Any state formed by low-lying states has a "kink" in the middle. To remove kink, need to keep almost all states.

Wilson suggested pert. thy. to fix it.

How to fix it (White and Noack, PRL **68**, 3487 (1992).)

One approach involves different boundary conditions.

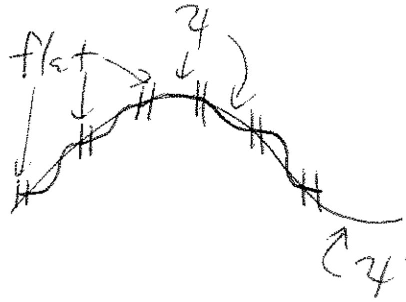
Periodic BCs? Only slightly better. Get "staircases" in excited states.



$\psi(0) = \psi(L)$

Free BCs? (Slope vanishes at edges.) Again, only slightly better (flat spots).

$$\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & \\ & & & \ddots \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & \\ & & & \ddots \end{pmatrix}$$



One solution: Combine states from different BCs.

- States must be orthogonalized.

Example: Fixed-Free combination.

Use $m/4$ states from each of Free-Free, Fixed-Free, Free-Fixed, Fixed-Fixed BC's. (4 diagonalizations each iteration)

$$\tilde{O} = \begin{pmatrix} \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ V_1^{ff} & \cdots & V_{m/4}^{ff} & V_1^{fo} & \cdots & V_{m/4}^{fo} & V_1^{of} & \cdots & V_{m/4}^{of} & V_1^{oo} & \cdots & V_{m/4}^{oo} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \end{pmatrix}$$

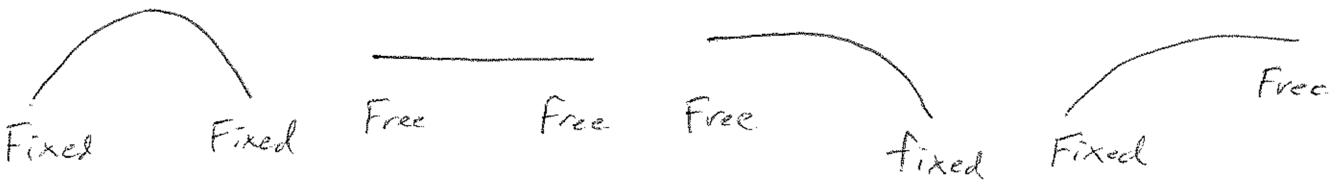
$O = \text{Gram-Schmidt}(\tilde{O})$ (Otherwise procedure is identical)

Test case: $m = 8$ states, 10 blockings:

	Exact	Standard RG	Fixed-Free
E_0	2.3508×10^{-6}	1.9207×10^{-2}	2.3508×10^{-6}
E_1	9.4032×10^{-6}	1.9209×10^{-2}	9.4032×10^{-6}
E_2	2.1157×10^{-5}	1.9214×10^{-2}	2.1157×10^{-5}
E_3	3.7613×10^{-5}	1.9217×10^{-2}	3.7613×10^{-5}

Results correct to 10 or 12 places.

Ground states:

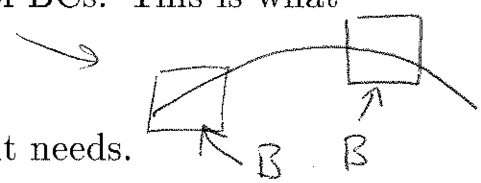


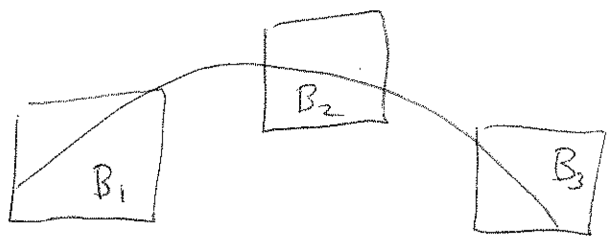
Other variations: periodic-antiperiodic works almost as well.

<u>Why does varying the boundary conditions work?</u>

When you isolate a block, that applies a particular BC to the block. The rest of the lattice, if it were there, would apply different BCs, so the states you keep aren't appropriate. You have two ways to rectify this. All the global features of the wfns end up applying BC's to the block.

1. Make each block able to represent a variety of BCs. This is what we just did with the fixed-free method.
   "Complete set" of BC's
2. Design each block to represent the exact BC it needs.





For noninteracting system, to get one state, need only <u>one state per block!</u>

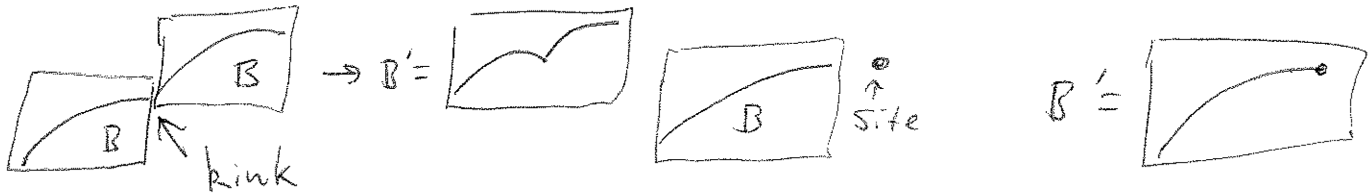
In method 2, block must know where it goes. Clearly method 2 must be iterative.

Method 1 doesn't work well for interacting systems–need too many states to represent response to lots of possible BCs. Also, it's not clear how to choose to vary the BC's in interacting systems. I tried several methods for Heisenberg chains—none worked.

Method 2 can be done in principle by diag. <u>whole</u> lattice, project out part of ~~state~~ ψ_G in B_1 for B_1 to keep.

How do we implement method 2?

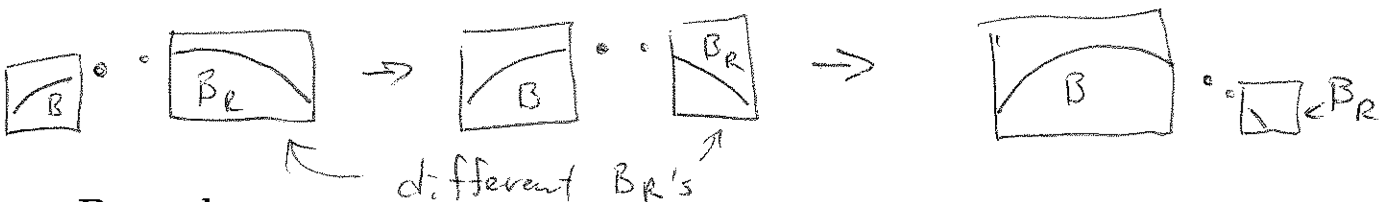
Can't add two blocks together–the blocks are appropriate for one location only. Must add a site onto a block: sites go anywhere.



Need right environment for block.

Need representation of rest of system. Use one block for left half of system, another for right half, with a few sites in between.

How do we choose the m states to keep? Diagonalize full system, and project out the left part of the m lowest energy eigenstates. Just like when we vary the BCs, we need to orthogonalize these left parts of the eigenstates.



Procedure

Start with a set of $\ell = 1 \ldots L$ approximate blocks representing the right-hand set of sites from ℓ to L.

Progressively add sites to left block. The left block grows, and we use progressively smaller right-hand blocks. (There is no way to "shrink" the right-hand block.) Store each left block as it is formed.

When you get to the right side, turn around and use the stored left blocks, adding sites to the right block.

Iterate until converged.

"Zipper"

DMRG for 1D particle in a box

Procedure

Initially $H = (2)$ and $T = (-1) =$ column vector.
We have a set of blocks H^R of all sizes up to L.

1. Form H for the whole lattice.

$$\bar{H} = \begin{array}{c} \begin{array}{cccc} m & 1 & 1 & m \end{array} \\ \left(\begin{array}{c|c|c|c} H_\ell & T_\ell & & 0 \\ \hline T_\ell^T & 2 & -1 & \\ \hline & -1 & 2 & T^R_{\ell+3} \\ \hline 0 & & T^{RT}_{\ell+3} & H^R_{\ell+3} \end{array}\right) \end{array}$$

2. Diagonalize $\bar{H}$, getting eigenvectors V^ℓ, $\ell = 1, \ldots, m$. Discard V^ℓ, $\ell = m+1, \ldots, N$.

3. Form matrix O

$$\tilde{O} = \begin{array}{c} \leftarrow m \rightarrow \\ \begin{pmatrix} V_1^1 & \cdots & V_1^m \\ \vdots & & \vdots \\ V_{m+1}^1 & \cdots & V_{m+1}^m \end{pmatrix} \end{array} \begin{array}{c} \uparrow \\ m+1 \\ \downarrow \end{array} \quad O = \text{Gram-Schmidt}(\tilde{O})$$

4. Change basis and truncate:

$$\tilde{H} = \begin{array}{c} \leftarrow m+1 \rightarrow \\ \left(\begin{array}{c|c} H_\ell & T_\ell \\ \hline T_\ell^T & 2 \end{array}\right) \end{array} \begin{array}{c} \uparrow \\ m+1 \\ \downarrow \end{array} \quad \tilde{T} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix} \begin{array}{c} \uparrow \\ m+1 \\ \downarrow \end{array}$$

$$H_{\ell+1} = O^T \tilde{H} O \qquad T_{\ell+1} = O^T \tilde{T}$$

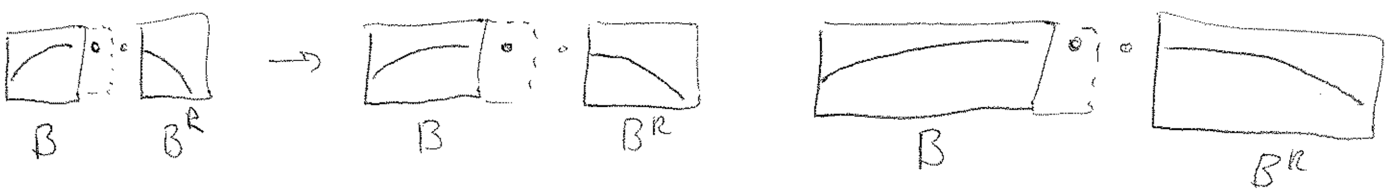
5. Iterate $\ell = 1$ to $\ell = L - 3$.

6. Reverse directions and go right to left.

7. Repeat until converged.

How do we get an initial set of approximate blocks to use at first?

At the beginning, when the left block is small, the best approximation we have for big right block is the biggest left block we have so far, but flipped around. $B^R = \text{reverse}(B)$

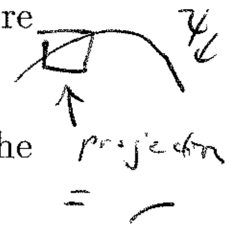


So in the first pass through the lattice, when H^R is called for in constructing $\bar{H}$, we just reverse the rows and columns of H to get it.

Interacting Systems

Now we want to generalize to interacting systems. This primarily consists of adding sites with an $\otimes$, not an $\oplus$.

Most of the DMRG procedure outlined before needs little change. The main question:
How do we project out a state for a block from a state of the entire lattice? Problem: the projection is many-valued.



Let $|i\rangle$ be the states of the block, and $|j\rangle$ be the states of the rest of the lattice. A state of the entire lattice can be written as

$$|\psi\rangle = \sum_{ij} \psi_{ij} |i\rangle |j\rangle$$

In general, there is no way to pick states $|\tilde{i}\rangle$ and $|\tilde{j}\rangle$ so that

$$|\psi\rangle = |\tilde{i}\rangle |\tilde{j}\rangle$$

Example: if the block has an average of N particles, it can still fluctuate into states with $N \pm 1$, $N \pm 2$, particles. Need at least one state for each number of particles. (A state without a definite N, such as the BCS wavefunction, doesn't help, either.)

We will need an *approximate* projection. What is the best projection? It comes from the density matrix.

Density Matrices

Reference: R.P. Feynman, *Statistical Mechanics: A Set of Lectures*

Let $|i\rangle$ be the states of the block (the *system*), and $|j\rangle$ be the states of the rest of the lattice (the rest of the *universe*). If ψ is a state of the entire lattice,

$$|\psi\rangle = \sum_{ij} \psi_{ij} |i\rangle |j\rangle$$

The reduced density matrix for the system is

$$\rho_{ii'} = \sum_j \psi^*_{ij} \psi_{i'j}$$

An operator A which acts only on the system can be written as

$$A = \sum_{ii'j} A_{ii'} |\cancel{\phi_i}\rangle |\cancel{\phi}_j\rangle \langle \cancel{\phi}_j| \langle \cancel{\phi}_{i'}| = \sum_{ii'} A_{ii'} |\cancel{\phi}_i\rangle \langle \cancel{\phi}_{i'}| \otimes \mathbf{1}_j$$

The expectation value of A can be written in terms of the density matrix

$$\langle A \rangle = \sum_{ii'j} A_{ii'} \psi^*_{ij} \psi_{i'j} = \sum_{ii'} A_{ii'} \rho_{i'i} = \mathrm{Tr} \rho A$$

A nice way of representing ρ is through its eigenstates $|v_\alpha\rangle$ and eigenvalues $w_\alpha \geq 0$ $(\sum_\alpha w_\alpha = 1)$

$\sum_\alpha w_\alpha = \mathrm{Tr}\,\rho = \mathrm{Tr}[\psi^*_{ij}\psi] = \sum_{ij} |\psi_{ij}|^2 = 1$ by normaliz.

$$\rho = \sum_\alpha w_\alpha |v_\alpha\rangle \langle v_\alpha|$$

show $v\rho v \geq 0 = \sum_{ii'j} v_i^* \psi^*_{ij} \psi_{i'j} v_{i'}$

w_α = prob of system being in state v_α

The $|v_\alpha\rangle$ provide the best way to project out important states of the block. We can argue several ways. Notice that

$$\langle A \rangle = \sum_\alpha w_\alpha \langle v_\alpha | A | v_\alpha \rangle$$

If for a particular α, $w_\alpha \approx 0$, we make no error in $\langle A \rangle$ if we discard $|v_\alpha\rangle$.

Thus projection with density matrix: diag ρ, keep m most probable eigenvalues w_α

Entanglement

Entanglement is a property of a state ψ divided into 2 parts — how quantum-correlated are the two parts?

Example: Two $S=\frac{1}{2}$'s. Q. Which state is more ~~cor~~ entangled?

(a) $|\uparrow\downarrow\rangle + |\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle + |\downarrow\uparrow\rangle$

(b) $|\uparrow\uparrow\rangle + |\downarrow\downarrow\rangle$

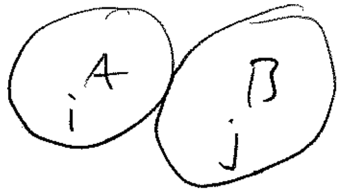
Answer: (b) is perfectly entangled.

(a) is unentangled

$$(|\uparrow\rangle + |\downarrow\rangle) \otimes (|\uparrow\rangle + |\downarrow\rangle)$$ product state

or $|x\text{-}\uparrow\rangle \otimes |x\text{-}\uparrow\rangle$

In general, how do you tell?



$$|\psi\rangle = \sum_{ij} \psi_{ij} \, |i\rangle |j\rangle$$

ψ_{ij} like a matrix

Singular Value Decomposition — Matrix Factorization

- works for any matrix

$\boxed{\text{Numerical recipes}}$

$$\underset{m\times n}{\psi} = \underset{\substack{m\times m \\ \text{orthog}}}{U} \; \underset{\substack{m\times m \\ \text{diag}}}{D} \; \underset{m\times n}{\tilde{V}}$$

$m\times n$ — rows are orthonormal

D has diag els, ≥ 0 – singular values

QI: Ψ Schmidt-decomposition Schmidt numbers, vectors

Unentangled: only one sig. value $\neq 0$

Normalization: $\sum_{\alpha} \lambda_\alpha^2 = 1$ λ_α^2 = prob of state $|\alpha\rangle_L |\Phi_\alpha\rangle_R$

Density matrices:

$$\rho = \Psi \Psi^T = U D V (V^T D^T U^T)$$
$$= U D^2 U^T \text{ diag. form}$$

So $w_\alpha = \lambda_\alpha^2$ Density matrix idea same as Schmidt-decomp.

– DMRG is very natural from QI point of view

Matrix Product States

1^{st} transformation 

$$|\alpha_2\rangle = \sum_{\alpha_1} O_2[s_2]_{\alpha_2 \alpha_1} \, |s_2\rangle |\alpha_1\rangle \qquad |\alpha_1\rangle = |s_1\rangle$$

2^{nd}

$$|\alpha_3\rangle = \sum_{\alpha_2} O_3[s_3]_{\alpha_3 \alpha_2} \, |s_3\rangle |\alpha_2\rangle$$

$$= \sum_{\alpha_2 \alpha_1} O_3[s_3]_{\alpha_3 \alpha_2} \, O_2[s_2]_{\alpha_2 \alpha_1} \, |s_3\rangle |s_2\rangle |s_1\rangle$$

All the way across (at step $\square \cdot\cdot$)

$$|\psi\rangle = \sum_{\alpha_2 \cdots \alpha_{L}} O_{L-1}[s_L]_{\alpha_L \alpha_{L-1}} \cdots O[s_2]_{\alpha_2 s_1} \, |s_L \cdots s_1\rangle$$

This is a matrix product state:

$$\psi(s_1 \cdots s_L) = A_1[s_1] \cdots A_L[s_L] \quad \leftarrow \text{set of 2 matrices}$$

1^{st} + last A's = Vectors
rest = matrices

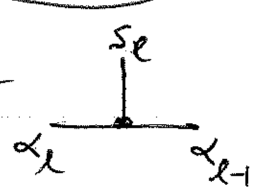
Ostlund & Rommer 1995

Specify $s_1 \cdots s_L$, Multiply matrices, get number – that is $\psi(s_1 \cdots s_L)$

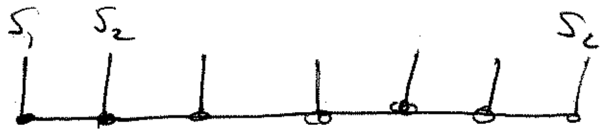
Another form

$$\psi(s_1 \cdots s_L) = \mathrm{Tr}\{ A_1[s_1] \cdots A_L[s_L] \}$$

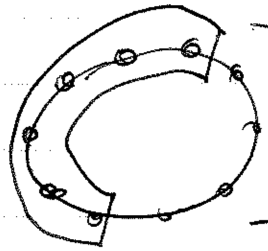
Diagrams

Let 

$\Longleftrightarrow \quad A[s_\ell]_{\alpha_\ell \alpha_{\ell-1}}$

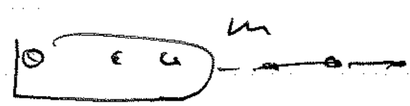
Then $|\psi\rangle \sim$ 

Natural ~~PBCs~~ Periodic BCs

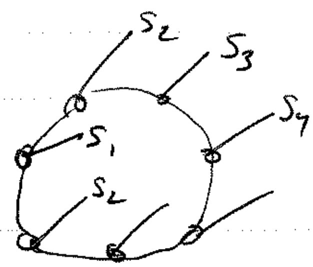


Block

entanglement at both ends

– If short range corrs, need m states for OBCs 

– Need m^2 for PBCs

Soln: 

$\psi(s_1 \ldots s_L) = \mathrm{Tr}\{A_1[s_1] \ldots A[s_L]\}$

all A's matrices, same size

Translational invariance: make all A's the same

$\psi = \mathrm{Tr}\{A[s_1] \ldots A[s_L]\}$ props of trace

$= \mathrm{Tr}\{A[s_2] \ldots A[s_1]\}$

Wilson, etc – Used MPS for calcs from beginning in NRG

History: Ostlund + Rommer – Crude Variational optimization

Verstraete ... good variational ~~meth~~ opt,

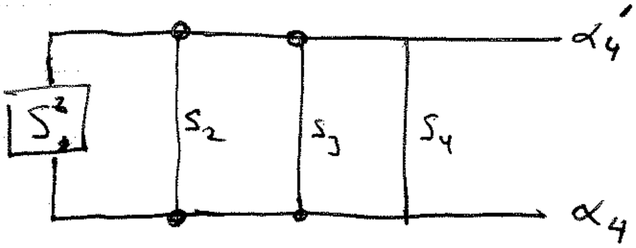
m like OBC, calc $\sim m^5$ like DMRG

Pippan, White, Evertz (2007) PBC with m^3

Diagrams for operators

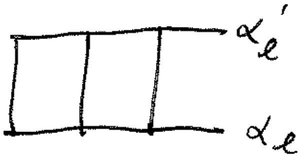
$$S_1^z \rightarrow \sum_{s_2} O^T[s_2] S_1^z O[s_2] \rightarrow \sum_{s_2 s_3} O^T[s_3] O^T[s_2] S_1^z O[s_2] O[s_3]$$

etc



$$(S_1^z)_{\alpha_4 \alpha_4'}$$

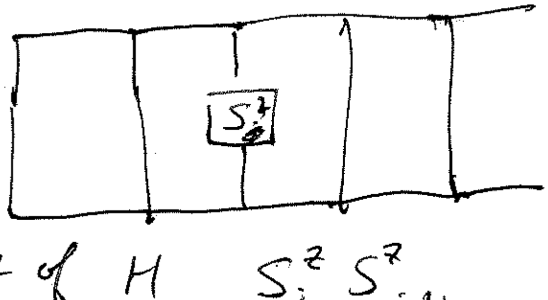
Note:



$= \delta_{\alpha_\ell \alpha_\ell'}$: D.M. eigenstates orthonormal

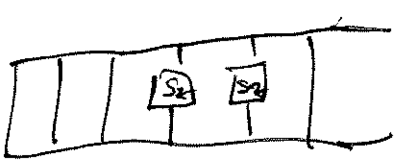
Comes from $\displaystyle\sum_{\alpha_{\ell-1} s_\ell} O[s_\ell]_{\alpha_\ell \alpha_{\ell-1}} O[s_\ell]_{\alpha_\ell' \alpha_{\ell-1}} = \delta_{\alpha_\ell \alpha_\ell'}$

S_j^z at step ℓ



part of H $\quad S_j^z S_{j+1}^z$

H block op = sum of terms like this



Efficient Programs

$\boxed{\alpha_\ell} - \underset{\ell+1}{\circ} - \underset{\ell+2}{\circ} - \boxed{\alpha_{\ell+3}} \approx \boxed{\alpha} - \boxed{\beta} \qquad \psi_{\alpha\beta}$

$$\tilde{H} = \tilde{H}_{\alpha\beta\alpha'\beta'}$$

Diag this H? $m^2 \times m^2$, m^6 calc time

Lanczos / Davidson sparse iterative: just need to do $H\psi$

$$m^2 \Big(\overset{m^2}{\quad}\Big)\Big(m^2\Big) \qquad m^4 \;? \qquad m^4 \text{ storage better}$$

Best

$$H = \sum_\gamma A^\gamma_{\alpha\alpha'} B^\gamma_{\beta\beta'} \qquad \text{e.g. } \begin{matrix} A^1 = S^z_\ell & A^2 = S^+_\ell \\ B^1 = S^z_{\ell+1} & B^2 = S^-_{\ell+1} \end{matrix}$$

$$H\psi \to \sum_{\gamma} A^\gamma_{\alpha\alpha'} B^\gamma_{\beta\beta'} \psi_{\alpha'\beta'}$$

$$\to \sum_\gamma \underbrace{A^\gamma_{\alpha\alpha'} \psi_{\alpha'\beta}}_{X \text{ temp}} B^\gamma_{\beta\beta'} \qquad \text{matrix mult}$$

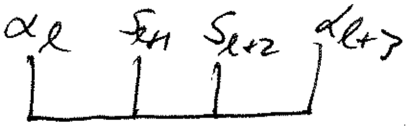
then XB

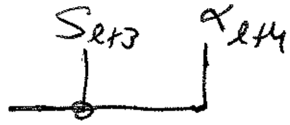
Calc time, storage $\sim m^3$

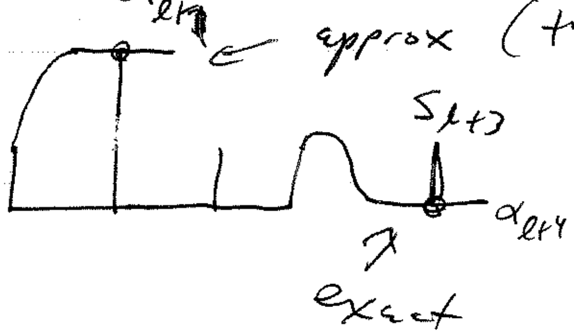
total $\sim L m^3$

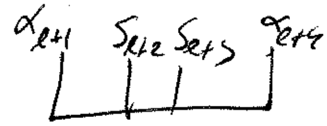
<u>hidden</u>: # of Lanczos iterations ~ 20-50

~~Wafn~~ Wavefn transformations — White (ref)

$\psi_{\alpha_\ell s_{\ell+1} s_{\ell+2} \alpha_{\ell+3}} \to$ 

$\alpha_{\ell+3} =$ 

$\psi \to$ 

$=$ 

So Apply two O's on left, right translates ψ to new basis

$\Rightarrow$ very good guess for Lanczos startup

$\Rightarrow$ 20-50 its $\longrightarrow$ 2-4

Extrapolation

Σ = Truncation error = sum of density matrix eigs thrown away
= discarded weight

$\Sigma \sim (\Delta\Psi)^2$ $\quad \Delta\Psi$ = error in wfn.

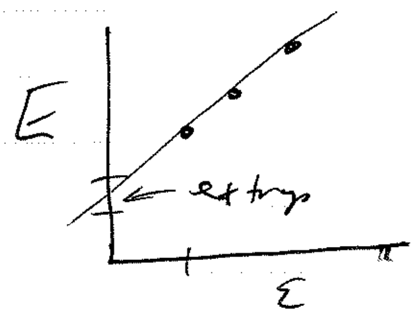
Variational method

$$E_{exact} = \langle \Psi_{ex} | H | \Psi_{ex} \rangle$$

$$E_{aprox} = \langle \Psi_{ex} + \Delta\Psi | H | \Psi_{ex} + \Delta\Psi \rangle$$

$$= \langle \Psi_{ex} | H | \Psi_{ex} \rangle + 2 \underbrace{\langle \Delta\Psi | H | \Psi_{ex} \rangle}_{E_{ex} \underbrace{\langle \Delta\Psi | \Psi_{ex} \rangle}_{0}} + \langle \Delta\Psi | H | \Delta\Psi \rangle$$

$$\Delta E \sim (\Delta\Psi)^2 \sim \Sigma$$



Observe excellent linearity

Set error $\sim \frac{1}{5} \cdot$ size of extrap

Improves energy by factor of ~5, plus error est.

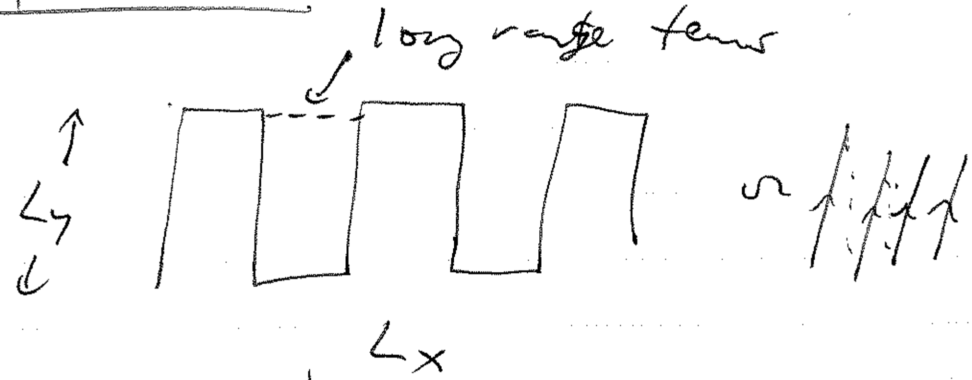
Other measurements

Expect $\Delta A \sim \Sigma^{\frac{1}{2}}$

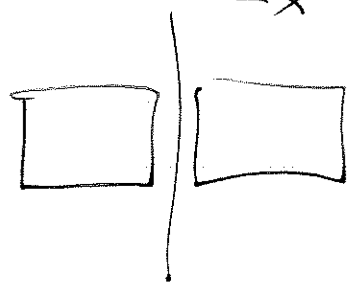
But: if A local, get $\Delta A \sim \Sigma$, extrap works
See White + Chen PRL 2007

Methods for 2D

TUScm



QI **Area Law**



S = von Neuman entropy

$$= \sum_\alpha -w_\alpha \ln w_\alpha \propto \text{area of cut}$$

Thus expect $S \sim L_y$ $\qquad S \sim \langle -\ln w_\alpha \rangle$

$\Rightarrow m \sim e^{aL_y}$ $\qquad w_\alpha \sim e^{-S} \sim \frac{1}{m}$

This looks hopeless, but a not so large (very model dependent)

Thus $L_y \sim 10$ ok, spin systems

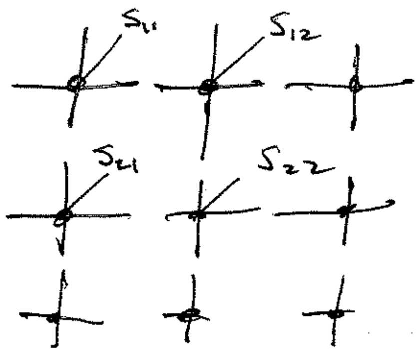
$L_y \sim 8$ t-J model

$L_y \sim 6$ Hubbard model

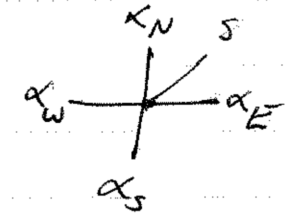
Verstraete, & Cirac, ..

PEPS projected entangled pair states

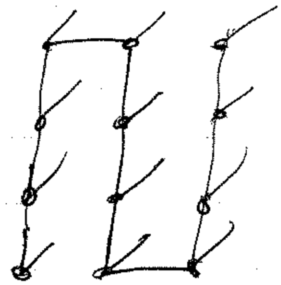
Much more natural 2D state



Basic unit tensor



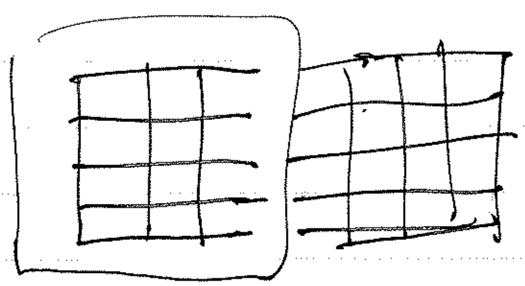
versus



Expect $m \sim 10$ to work great — but don't know

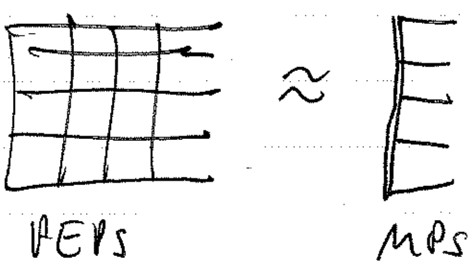
Problem: 1) How to evaluate E + props?
2) How to optimize tensors?

Contraction to evaluate $\psi(s_1 \cdots s_L)$



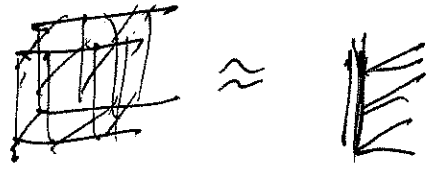
partial contraction: e^{aL_y} degrees of freedom!

Soln



Fit MPS to PEPS left block

Operators – two layers



Do fit for each op or edge, + H ~~east~~ block

Calc time for fitting, etc: sq lattice $\sim m^{10}$

Total calc time: $\sim L_x L_y m^{10} \cdot$ lots of iterations

versus (TVScan)

$$L_x \underbrace{L_y}_{\text{or } L_y^2} \left(e^{aL_y}\right)^3 \quad \text{or} \quad L_x L_y^2 e^{3aL_y}$$

Which is better?

Asymptotically: Peps

Now : TVScan