

# Wave Function Optimization Algorithms in QMCPACK



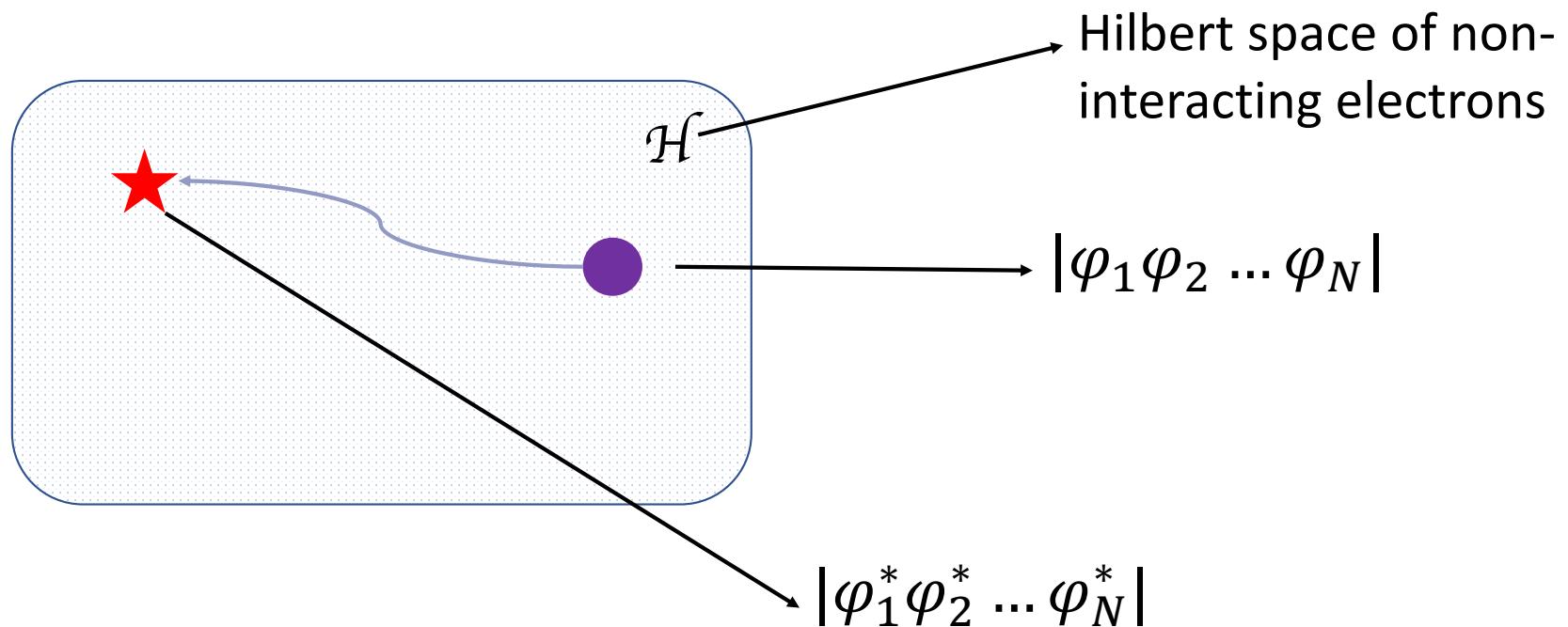
Berkeley  
UNIVERSITY OF CALIFORNIA

Luning Zhao

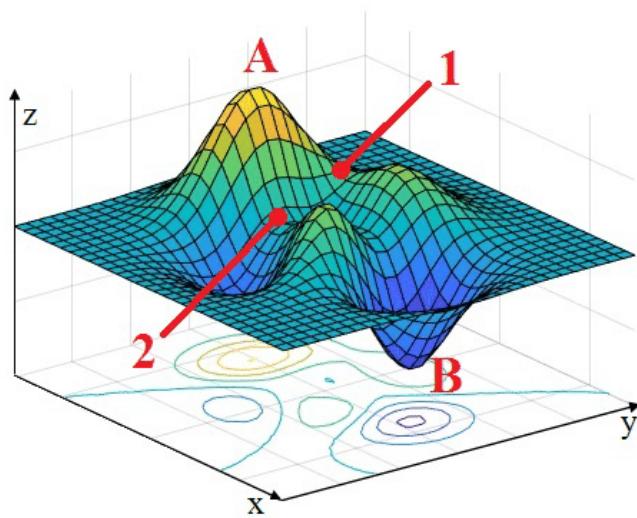
University of California, Berkeley

2019.05.15

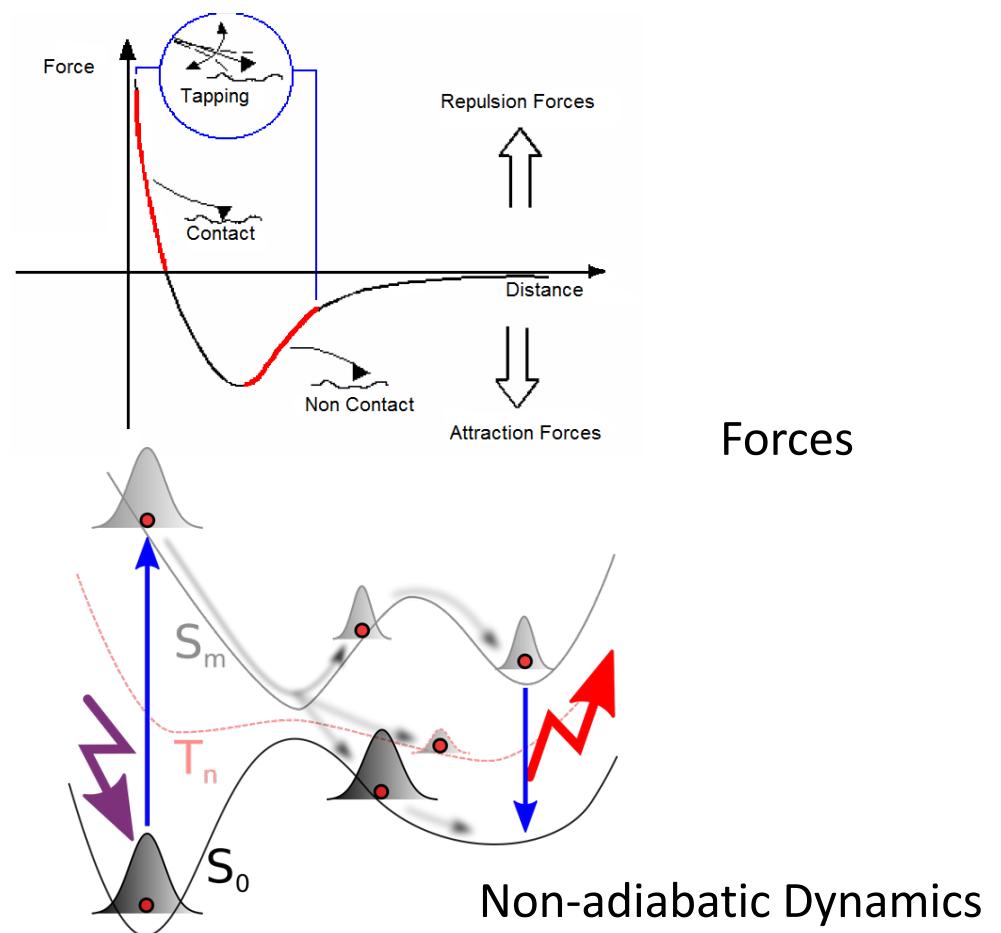
# The Motivation of Wave Function Optimization



# What can we compute with better wave functions?



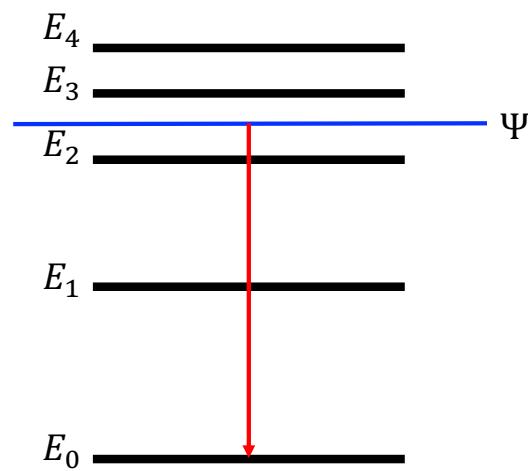
Potential Energy Surfaces:  
Barrier Height, Excitation  
Energy, Reaction Rate



# Variational Principle For Ground and Excited State

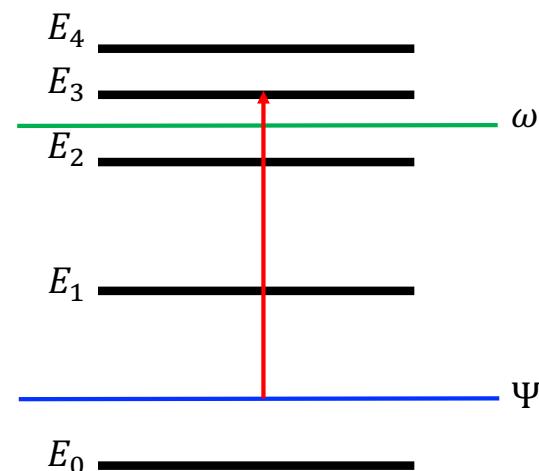
Ground State

$$E[\Psi] = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq E_0$$

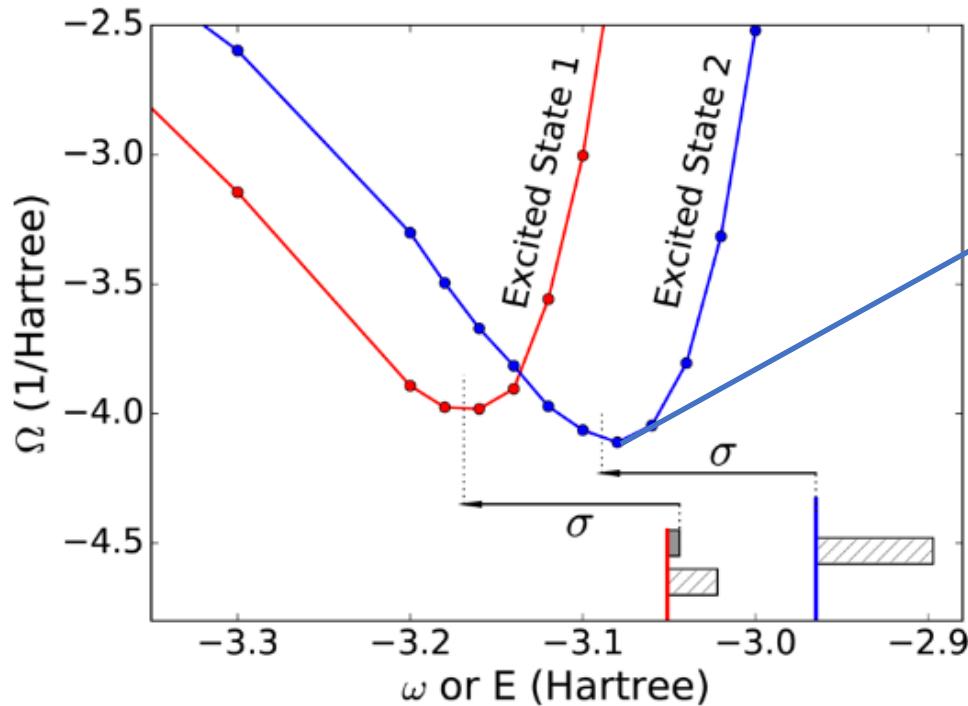


Excited State

$$\Omega[\Psi] = \frac{\langle \Psi | \omega - H | \Psi \rangle}{\langle \Psi | (\omega - H)^2 | \Psi \rangle}$$



# How to choose $\omega$ ?



Rule of Thumb: Take  $\omega$  as the initial energy subtracted by its standard deviation.

# Optimization Method in QMCPACK: The Linear Method

Wave Function with Parameters  $\vec{p}$ :  $\Psi(\vec{p})$

First Order Taylor Expansion around  $\vec{p}$ :

$$\Psi(\vec{p} + \Delta\vec{p}) = \Psi(\vec{p}) + \sum_{i=1} \frac{\partial\Psi}{\partial p_i} \Delta p_i = \sum_{i=0} \frac{\partial\Psi}{\partial p_i} \Delta p_i$$

Plug  $\Psi(\vec{p} + \Delta\vec{p})$  in  $E[\Psi]$  or  $\Omega[\Psi]$  :

$$E[\Psi(\vec{p} + \Delta\vec{p})] = \frac{\sum_{ij} \left\langle \frac{\partial\Psi}{\partial p_i} \middle| H \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle}{\sum_{ij} \left\langle \frac{\partial\Psi}{\partial p_i} \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle}$$

$$\Omega[\Psi(\vec{p} + \Delta\vec{p})] = \frac{\sum_{ij} \left\langle \frac{\partial\Psi}{\partial p_i} \middle| \omega - H \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle}{\sum_{ij} \left\langle \frac{\partial\Psi}{\partial p_i} \middle| (\omega - H)^2 \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle}$$

# Optimization Method in QMCPACK: The Linear Method

Taking derivatives w.r.t.  $\Delta\vec{p}$  and set the derivatives to 0:

$$\sum_j \left\langle \frac{\partial\Psi}{\partial p_i} \middle| H \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle \Delta p_j = E \sum_j \left\langle \frac{\partial\Psi}{\partial p_i} \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle \Delta p_j$$

$$\sum_j \left\langle \frac{\partial\Psi}{\partial p_i} \middle| \omega - H \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle \Delta p_j = \Omega \sum_j \left\langle \frac{\partial\Psi}{\partial p_i} \middle| (\omega - H)^2 \middle| \frac{\partial\Psi}{\partial p_j} \right\rangle \Delta p_j$$

Both equations are in the form of a generalized eigenvalue problem:

$$\tilde{H}\Delta\vec{p} = \lambda\tilde{S}\Delta\vec{p}$$

# Optimization Method in QMCPACK: The Linear Method

Perform Inverse Taylor expansion:

$$p_i = p_i + \Delta p_i$$

1. Build  $\tilde{H}$  and  $\tilde{S}$  matrices.
  2. Solve the Generalized Eigenvalue Problem.
  3. Update Wave Function Parameters.
  4. If not converged, go to step 1.
- 
- } One Step in LM

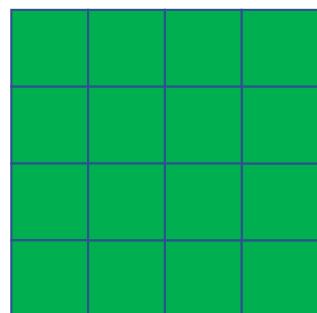
# Stabilizing Linear Method: Step Size Control

Modify the Hamiltonian:

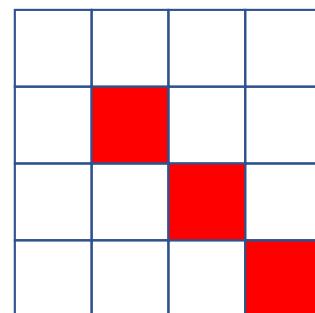
$$\tilde{H} = \tilde{H} + \alpha A + \beta B$$

Diagonal Shift Matrix  $A$ :

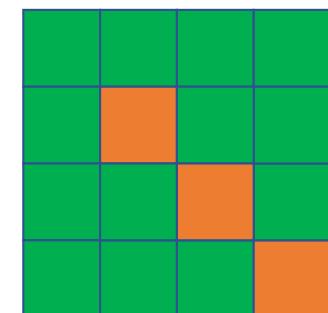
$$A_{xy} = \delta_{xy}(1 - \delta_{x0})$$



+



=



# Stabilizing Linear Method: Step Size Control

Projection Matrix  $Q$ : Project the current wave function out

$$Q_{xy} = \delta_{xy} - \delta_{x0}(1 - \delta_{y0})S_{0y}$$

Shift Matrix  $T$ : Penalize directions with larger norms

$$T_{xy} = (1 - \delta_{x0}\delta_{y0})[Q^\dagger S Q]_{xy}$$

Shift Matrix  $B$ : Transform back to the original basis

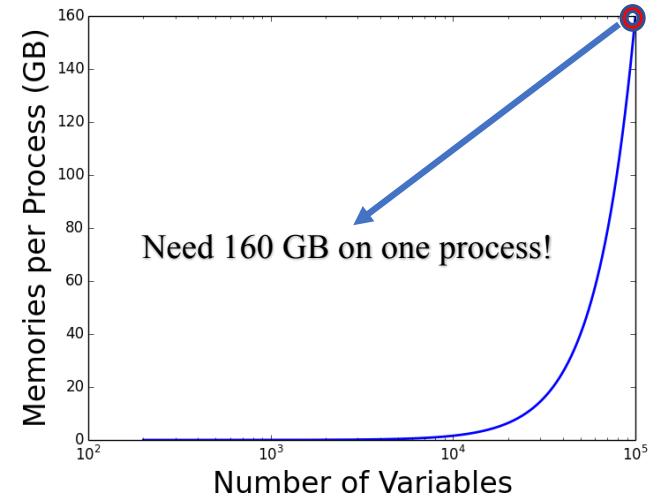
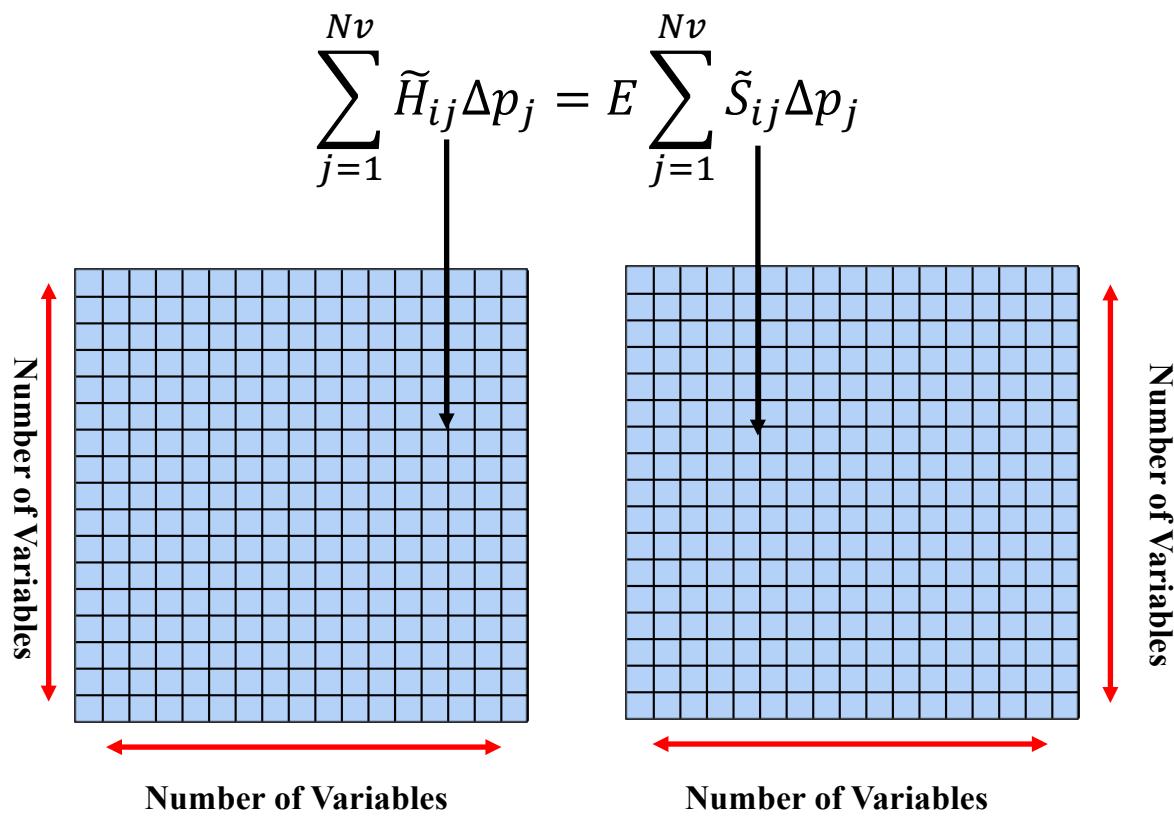
$$B = (Q^\dagger)^{-1} T Q^{-1}$$

# How to Choose the Shift Value?

adaptive:

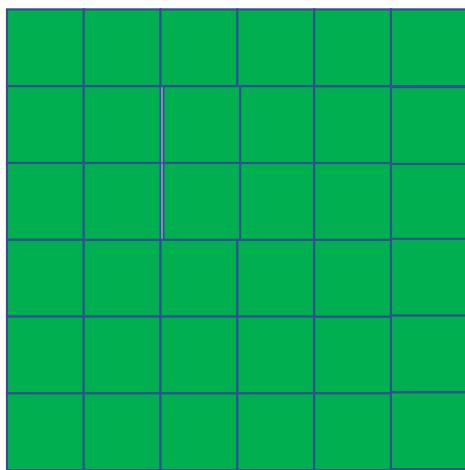
1. Set initial  $\alpha$  and  $\beta$
2. Small shifts:  $\alpha/4, \beta/4$ , Medium Shifts:  $\alpha, \beta$ , Large Shifts:  $4\alpha, 4\beta$
3. Solve the generalized eigenvalue problem for these three sets of shifts.
4. Use correlated sampling to compare the updated wave function based on three shifts
5. Small shifts wins: lower the shifts Medium shifts wins: no change to the shifts Large shifts wins: increase the shifts

# Memory Bottleneck of LM

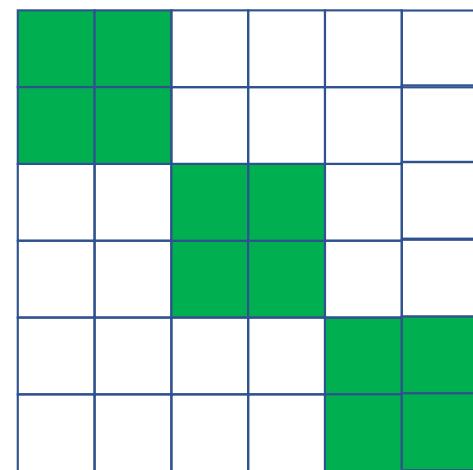


A few GB of memory

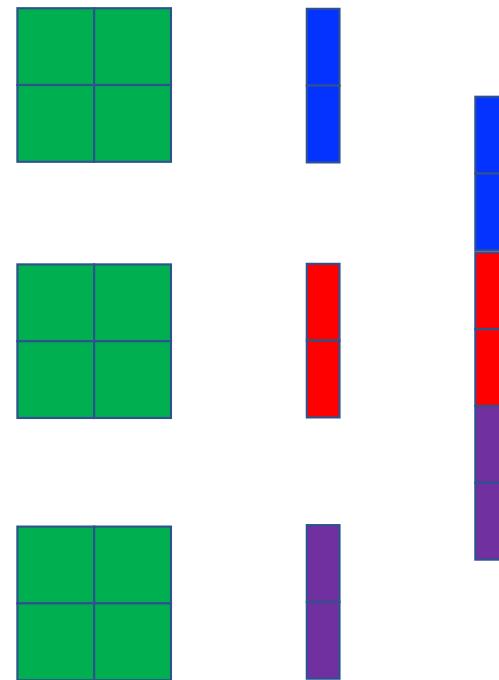
# Blocked LM: A Crude Idea



Initial Dense Matrix

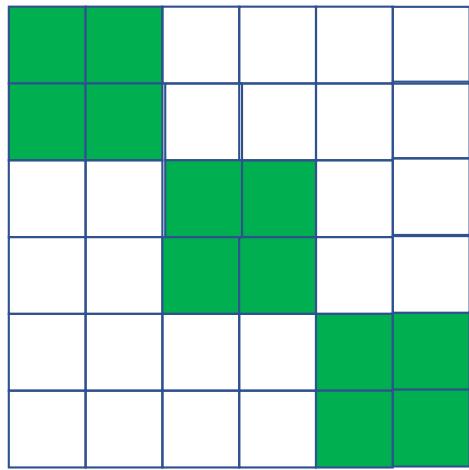


Divide Variables into  
Blocks

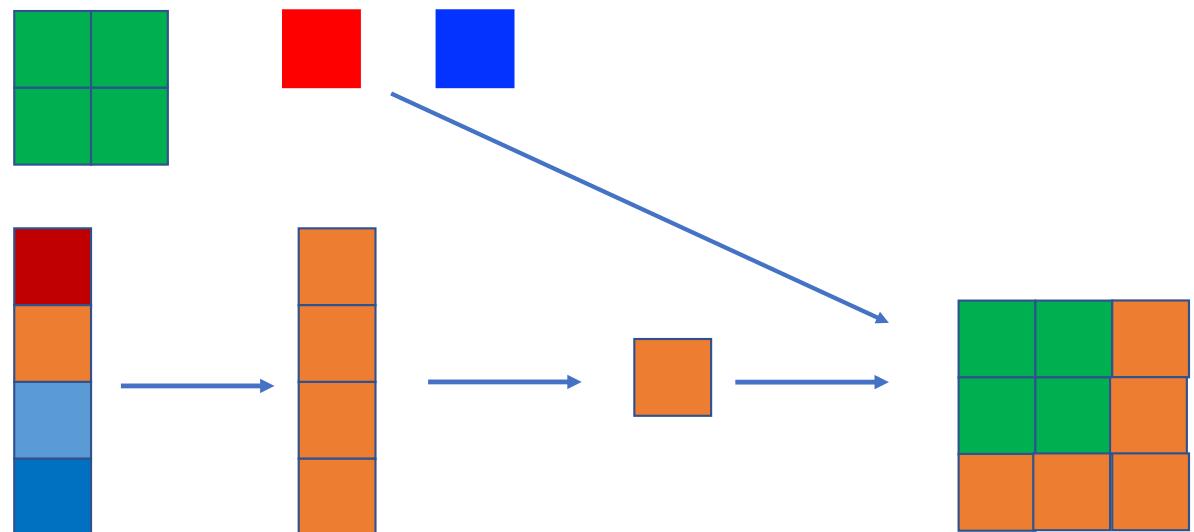


Solve for each Block

# Blocked LM: Account for Inter-Block Coupling



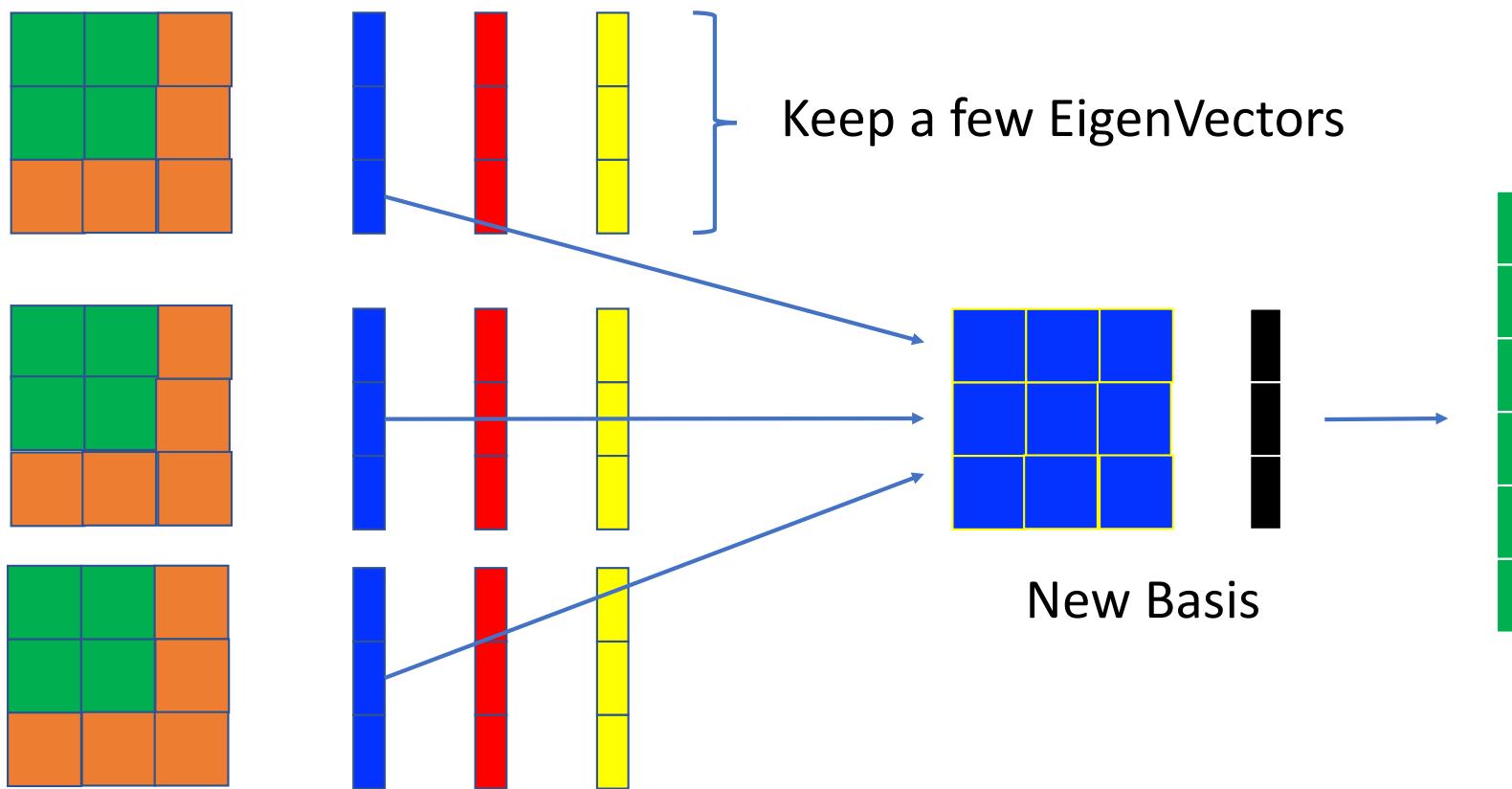
Uncorrelated Blocks



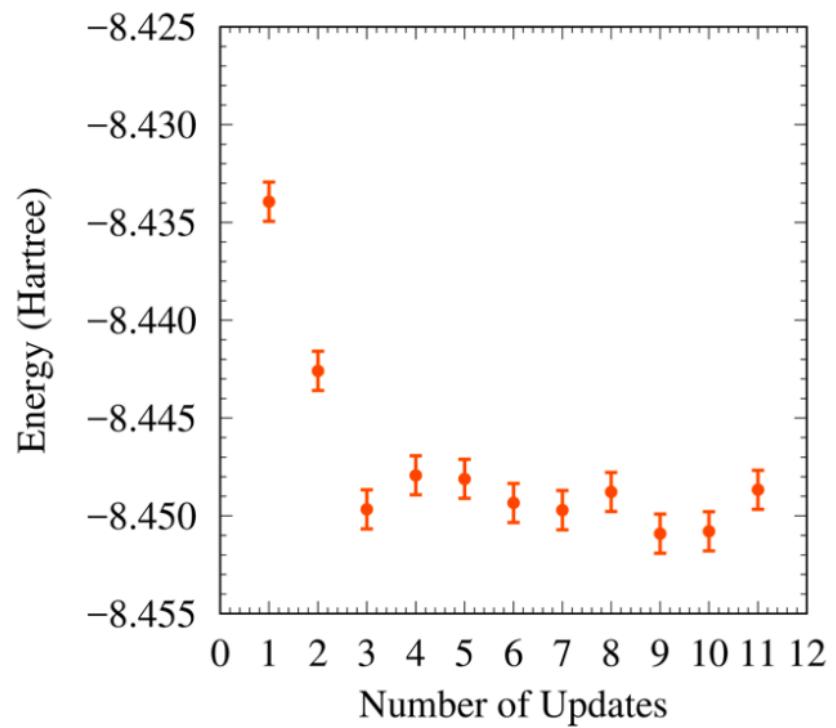
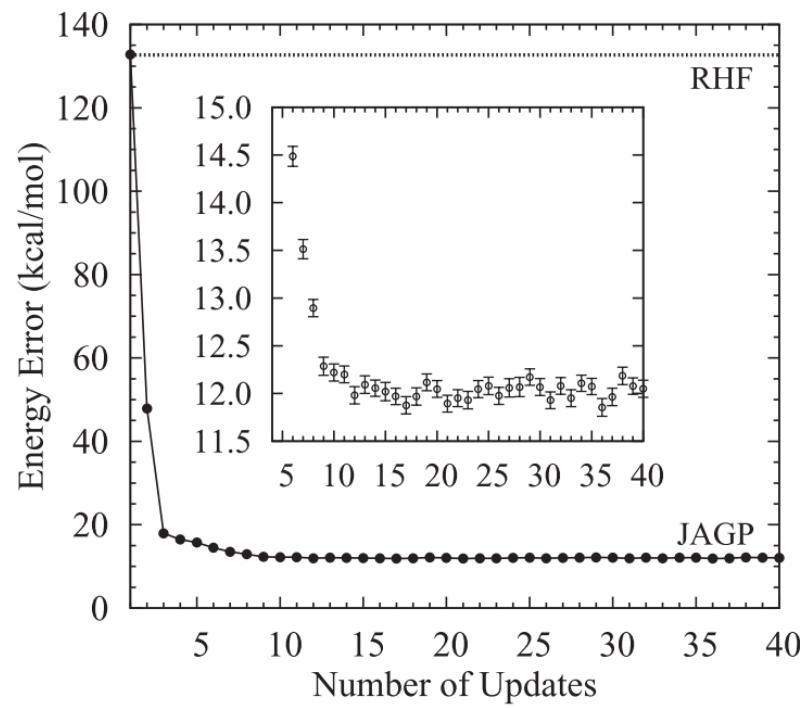
Old Update Directions

Correlated Blocks

# Blocked LM: Account for Inter-Block Coupling



# How does it work in reality?



# How to use LM/BLM in QMCPACK

```
<loop max="10">  
  <qmc method="linear" move="pbyp" checkpoint="-1" gpu="no">  
    <parameter name="MinMethod">adaptive</parameter>  
    <parameter name="targetExcited"> yes </parameter>  
    <parameter name="omega"> -17.779 </parameter>  
    .....  
  </qmc>  
</loop>
```

Specify linear method

Adapt shifts  
automatically

# Parameters of LM in QMCPACK

1. targetExcited: yes/no: whether to do target function/energy minimization
2. omega: real number: Energy shift used to target different excited states
3. shift\_i: 1.0: value of the identity shift
4. shift\_s: 1.0: value of the overlap shift
5. max\_param\_change: 0.3: the maximum allowed parameter update
6. chase\_lowest: yes/no: whether to chase the lowest eigenvector in iterative solver
7. chase\_closest: yes/no: whether to chase the eigenvector that is closest to the initial guess

## Parameters of BLM in QMCPACK

1. block\_lm: yes/no: whether to switch to BLM
2. nblocks: integer: number of blocks in BLM
3. nkept: integer: number of eigenvectors to keep per block in BLM
4. nolds: integer: number of old update vectors used in BLM

Rule of Thumb: BLM should be turned on when the number of parameters > 5000.

Usually a few hundred blocks and a handful of nkept and nolds are sufficient to balance memory and accuracy.

# Type of Parameters that LM/BLM can optimize

Everything!

$$\Psi(\vec{p}) = e^J e^K \left( \sum_i C_i D_i(\vec{b}) \right)$$

The diagram illustrates the components of the wavefunction  $\Psi(\vec{p})$ . The expression is shown as  $\Psi(\vec{p}) = e^J e^K \left( \sum_i C_i D_i(\vec{b}) \right)$ . Four blue arrows point from the text labels below to specific parts of the equation:

- A blue arrow points from "Jastrow Variable" to the term  $e^J$ .
- A blue arrow points from "Orbital Rotation Variable" to the term  $e^K$ .
- A blue arrow points from "CI Vector" to the term  $\sum_i C_i$ .
- A blue arrow points from "Backflow Variable" to the term  $D_i(\vec{b})$ .

# Output of LM optimizer I: Initial Wave Function Statistics

target function =	-0.372878390901
le_mean =	-16.328583204639
les_mean =	268.408473184878
stat err =	0.023008167982
autocorr =	5.928282759927
target nu stat err =	0.023008167982
target dn stat err =	0.098081779036
target stat err =	0.011108538005
std dev =	1.336388046461
variance =	1.785933010723

# Output of LM optimizer II: Davidson Solve for the Eigenvector

```
davidson iteration  0: krylov dim =  1  tar_fn = -0.372900676924 residual = 4.72e-01 smallest_sin_value = 1.00e+00
davidson iteration  1: krylov dim =  2  tar_fn = -0.403505269722 residual = 3.52e-01 smallest_sin_value = 1.00e+00
davidson iteration  2: krylov dim =  3  tar_fn = -0.416670210607 residual = 7.63e-01 smallest_sin_value = 1.00e+00
davidson iteration  3: krylov dim =  4  tar_fn = -0.436226679861 residual = 1.43e-01 smallest_sin_value = 1.00e+00
davidson iteration  4: krylov dim =  5  tar_fn = -0.455304517531 residual = 1.29e-01 smallest_sin_value = 1.00e+00
```

The largest weight on the derivative vector for shift  $2.5000\text{e-}01$  is  $8.966562\text{e-}01$

If this value is larger than the largest allowed value, this update will be rejected.

# Output of LM optimizer III: Correlated Sampling

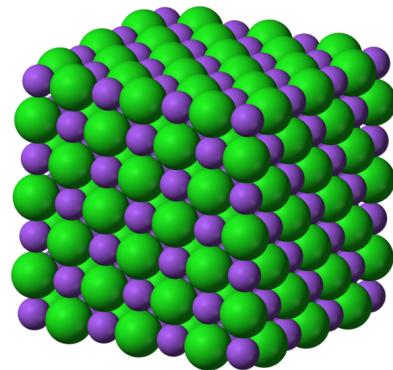
target function =	-0.383191837631
le_mean =	-16.356113520223
les_mean =	269.210850877312
stat err =	0.022583428126
autocorr =	6.041048391654
target nu stat err =	0.022583428126
target dn stat err =	0.091420194353
target stat err =	0.011225165065
std dev =	1.299417490715
variance =	1.688485815177

target function =	-0.172618968424	-0.560511869170	-0.405419204271
le_mean =	-16.466764668375	-16.577242846904	-16.393817912865
les_mean =	272.962774420655	275.504631302512	270.254981559569
stat err =	0.020657344609	0.012765289390	0.021096090120
autocorr =	4.719045657358	4.657879128558	5.942686406468
target nu stat err =	0.020657344609	0.012765289390	0.021096090120
target dn stat err =	0.080950566099	0.043639262866	0.082508798126
target stat err =	0.004801104087	0.012869472409	0.011575364722
std dev =	1.344814560945	0.836472283802	1.223842591564
variance =	1.808526203330	0.699685881570	1.497790688925

# Output of LM optimizer IV: Comparison of Different Shifts

shift_i	shift_s	max param change	cost function value	
N/A	N/A	N/A	-0.383191837631	initial
N/A	N/A	N/A	N/A	bad update
1.0000e+00	1.0000e+00	1.5141e-01	-0.560511869170	<--
4.0000e+00	4.0000e+00	3.3374e-02	-0.405419204271	

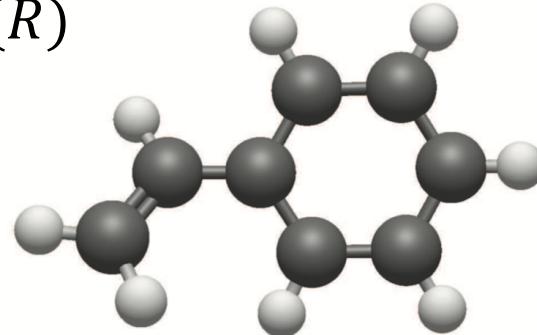
# Upcoming Optimization Features in QMCPACK



Complex Wave Function  
Parameters: Solid State

$$\Psi_G^2(R) = |\Psi(R)|^2 + \varepsilon \sum_i \Psi_i^2(R)$$
$$\Psi_G^2(R) = |\Psi(R)|^2 + \varepsilon |\nabla^2 \Psi(R)|$$

Guiding Function other than  
 $\Psi(R)$



Hybrid Optimization with  
BLM and Gradient Descent

# Examples in the Hands on Session

1. Energy Optimization of all-electron H<sub>4</sub> with Jastrow Multi-Slater Wave Function and Orbital Optimization.
2. Excited State Optimization of H<sub>2</sub>O with Jastrow Multi-Slater Wave Function and Orbital Optimization.
3. Blocked LM Optimization of C<sub>2</sub>.

# References

1. C. J. Umrigar and C. Filippi, Energy and Variance Optimization of Many-Body Wave Functions, *Phys. Rev. Lett.* 94, 150201 (2005)
2. L. Zhao and E. Neuscamman, An Efficient Variational Principle for the Direct Optimization of Excited States, *J. Chem. Theory Comput.* 12, 3436-3440 (2016)
3. L. Zhao and E. Neuscamman, A Blocked Linear Method for Optimizing Large Parameter Sets in Variational Monte Carlo, *J. Chem. Theory Comput.* 13, 2604-2611 (2017)
4. L. Otis and E. Neuscamman, Complementary First and Second Derivative Methods for Ansatz Optimization in Variational Monte Carlo, arXiv:1904.10087v1

# Acknowledgement

- Eric Neuscamman
- Graduate students and postdocs in Neuscamman group
- Collaborators in CPSFM



**QMCPACK** **CPSFM**