

# Orbital Optimization

1. Motivation
2. Implementation
3. Tutorial on  $C_2$

**Joshua P. Townsend & Amanda E. Duménil**  
**Sandia National Laboratories**  
**Albuquerque, NM 87185**

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

# Motivation

---

- **Slater-Jastrow (SJ) is a simple & common ansatz:**

$$\Psi^{SJ}(X) = e^{J(X)} \det \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_N) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \cdots & \phi_2(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\mathbf{x}_1) & \phi_N(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_N) \end{vmatrix}$$

- **Quality (variance) is set by nodal surface, i.e. by the  $\phi_k$ 's**
- **How to get 'good' orbitals? (especially where DFT struggles!)**
- ***Orbital optimization improves wave function quality with essentially no performance cost!***

# Implementation: Key Ideas

---

- Express each SPO as a linear combination in some space:

$$\tilde{\phi}_P = \sum_Q \phi_Q U_{QP}$$

**'Rotated' SPO** (blue arrow pointing to  $\tilde{\phi}_P$ )

**'bare' SPO** (red arrow pointing to  $\phi_Q$ )

**Unitary matrix** (black arrow pointing to  $U_{QP}$ )

- In QMCPACK,  $U$  is built from an anti-hermitian matrix,  $\kappa$  :

$$U = e^{\kappa} \quad \kappa = -\kappa^\dagger$$

# Implementation: Key Ideas

- Elements of kappa matrix are related to single excitations<sup>1,2,3</sup>:

$$\hat{K} = \sum_{ij} \kappa_{ij} \hat{E}_{ij}$$

↖  
Variational  
parameter

$$\hat{E}_{ij}^{\sigma} = \hat{a}_{i\sigma}^{\dagger} \hat{a}_{j\sigma}$$

Excitation from  $j \rightarrow i$  for spin channel  $\sigma$

- Action of  $\hat{E}_{ij}^{\sigma}$  on a determinant is to replace  $|\phi_j\rangle$  with  $|\phi_i\rangle$
- $\hat{K}$  has structure:

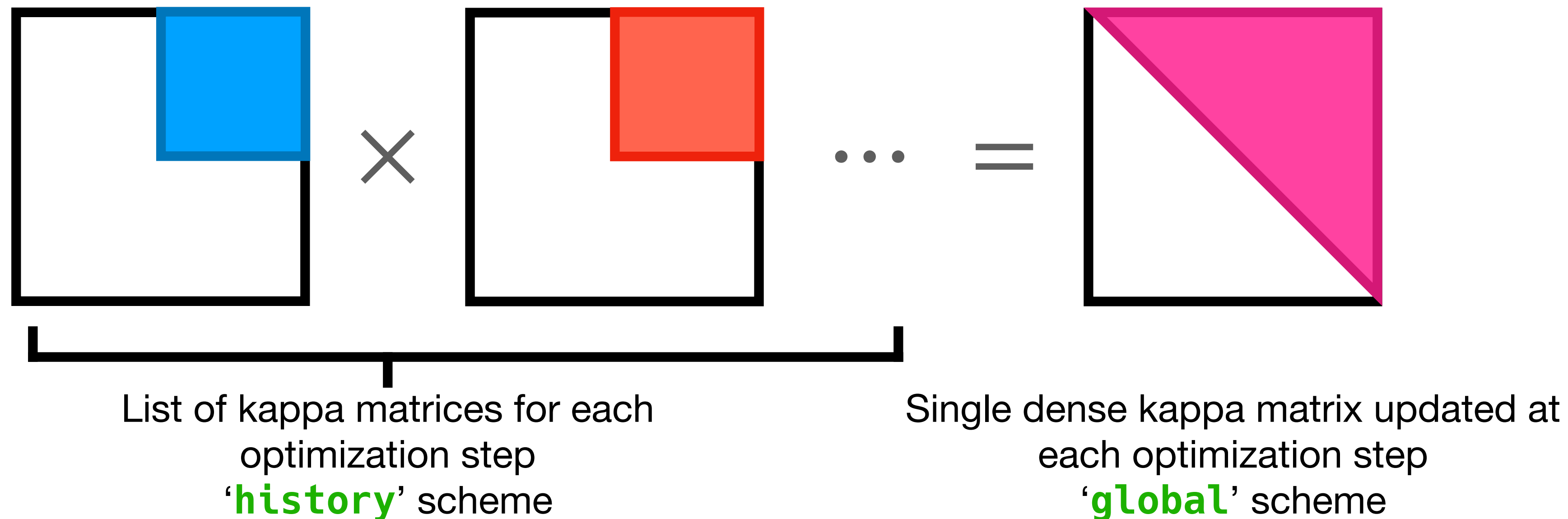
$$\kappa = \begin{array}{cc} & \begin{array}{c} \text{Nocc} \\ \text{Nvirt} \end{array} \\ \begin{array}{c} \text{Nocc} \\ \text{Nvirt} \end{array} & \begin{array}{|c|c|} \hline 0 & \text{Red Box} \\ \hline \text{Red Box} & 0 \\ \hline \end{array} \end{array}$$

←  $N_{occ} \times N_{virt}$  parameters

-c.c.

# Implementation: Key Ideas

- After optimization the net rotation from bare SPOs to optimal SPOs is dense:



- QMCPACK provides i/o for both schemes. Equivalent results but storage and start-up time are different. Global is default.

# Input Files: How to turn on orbopt

RotatedSPO wraps  
a bare SPOset

Determinant built  
from RotatedSPOs

```
<wavefunction name="psi0" target="e">
  <sposet_collection type="MolecularOrbital" href="c4q.orbs.h5" source="ion0">
    <basiset name="LCAOBSset" id="atomicBasisSet0" keyword="GT0" transform="no"> </basiset>
    <rotated_sposet name="rot_spo_up" method="global">
      <sposet name="spo_up" basiset="LCAOBSset" size="100">
        <coefficient spindataset="0"> </coefficient>
      </sposet>
    </rotated_sposet>
    <rotated_sposet name="rot_spo_down" method="global">
      <sposet name="spo_down" basiset="LCAOBSset" size="100">
        <coefficient spindataset="0"> </coefficient>
      </sposet>
    </rotated_sposet>
  </sposet_collection>
  <determinantset key="GT0" source="ion0" transform="no" type="M0">
    <slaterdeterminant>
      <determinant id="rot_spo_up"> </determinant>
      <determinant id="rot_spo_down"> </determinant>
    </slaterdeterminant>
  </determinantset>
</wavefunction>
```

Total number of  
SPO's  
(Must be > Nelecs)

Global: Store/apply a single net rotation  
History: Apply many small rotations



# Input Files: How to run optimized wfn

NB: Jastrow is ignored in this screenshot, but you should probably use one!

```
<wavefunction name="psi0" target="e">
  <sposet_collection type="MolecularOrbital" href="c4q.orbs.h5" source="ion0">
    <basisset name="LCAOBSset" id="atomicBasisSet0" keyword="GT0" transform="no"></basisset>

    <rotated_sposet name="rot_spo_up" method="global">
      <sposet name="spo_up" basisset="LCAOBSset" size="108">
        <coefficient spindataset="0"> </coefficient>
      </sposet>
    </rotated_sposet>
    <rotated_sposet name="rot_spo_down" method="global">
      <sposet name="spo_down" basisset="LCAOBSset" size="108">
        <coefficient spindataset="0"> </coefficient>
      </sposet>
    </rotated_sposet>
  </sposet_collection>

  <determinantset key="GT0" source="ion0" transform="no" type="M0">
    <slaterdeterminant>
      <determinant id="rot_spo_up">
        </determinant>
      <determinant id="rot_spo_down">
        </determinant>
      </slaterdeterminant>
    </determinantset>
    <override_variational_parameters href="opt.s032.vp.h5"/>
  </wavefunction>
```

Read in rotation  
params stored in  
vp.h5 file

Ensure correct path!

# Output Files: How to tell it's working (Optimization)

Check name



```
Single Slater determinant
```

```
-----
```

```
Determinant
```

```
-----
```

```
Name: rot_spo_up   Spin group: 0   SP0 name: rot_spo_up
```

```
Setting delay_rank to default value 1
```

```
Using rank-1 Sherman-Morrison Fahy update (SM1)
```

```
Running on CPU.
```

Check method



```
Orbital rotation using global rotation
```

```
nparams_active: 92 params2.size(): 0
```

Check params



```
Parameter name
```

```
Value
```

```
rot_spo_up_orb_rot_0000_0004    0.0000000e+00  0  1  ON  0
```

```
rot_spo_up_orb_rot_0000_0005    0.0000000e+00  0  1  ON  1
```

```
rot_spo_up_orb_rot_0000_0006    0.0000000e+00  0  1  ON  2
```

```
rot_spo_up_orb_rot_0000_0007    0.0000000e+00  0  1  ON  3
```



# Output Files: How to tell it's working (Production)

Loads in rotation  
parameters



```
udC_23      0.747054e-05      1      1      ON      23
udC_24     -1.487620e-06      1      1      ON      24
udC_25     -8.660394e-07      1      1      ON      25
```

```
Reading variational parameters from opt.s099.vp.h5
Adding psi0 TrialWaveFunction to the pool
```

```
Hamiltonian and observables
```

```
-----
```

```
Name: h0
```

```
QMCHamiltonian::addOperator Kinetic to H, physical Hamiltonian
```

```
Pseudo Potential
```

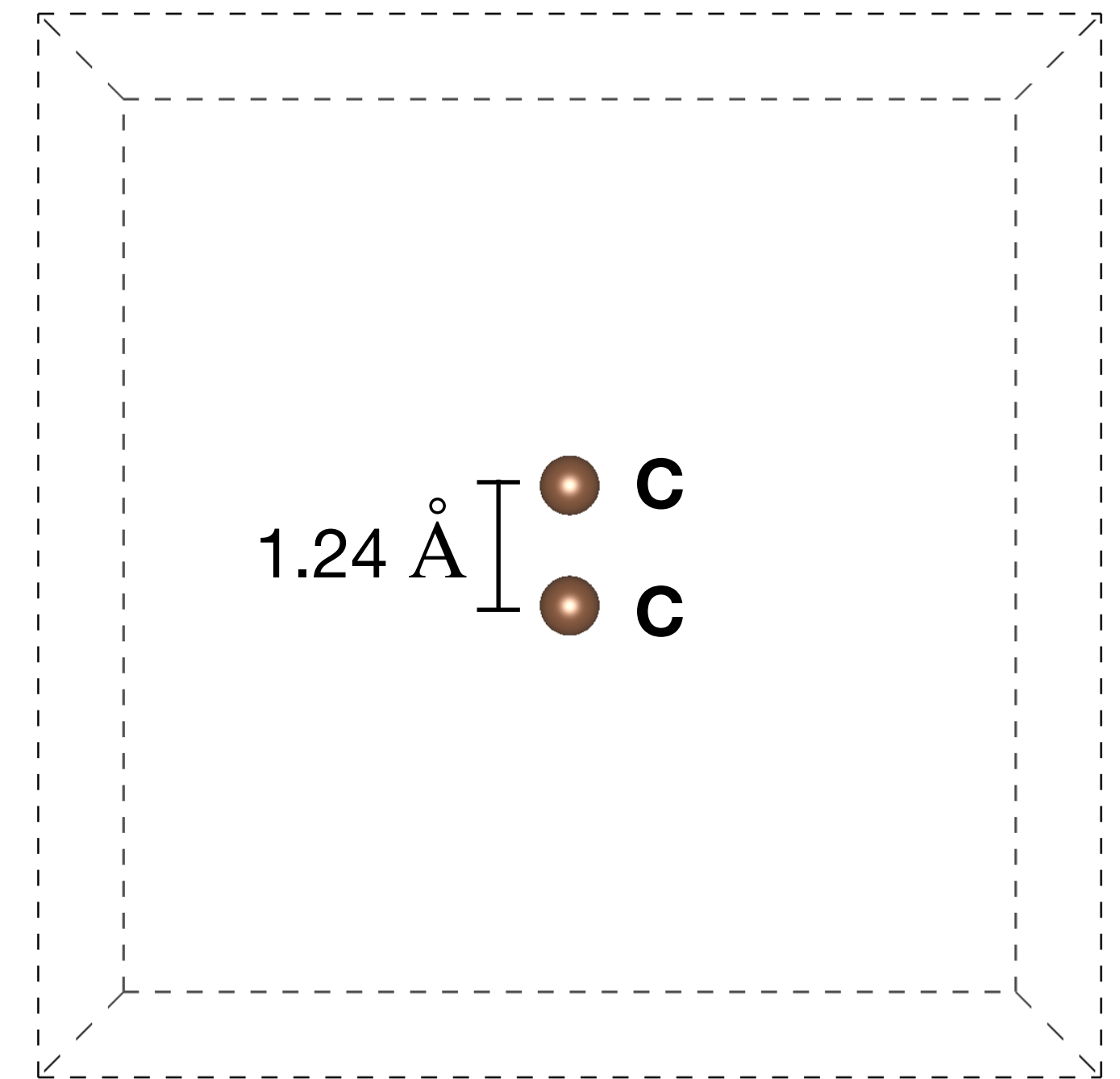
```
-----
```

```
Name: PseudoPot      Wavefunction : psi0
```

# Tutorial: Carbon dimer

---

- Clone: [https://github.com/QMCPACK/qmcpack\\_workshop\\_2023](https://github.com/QMCPACK/qmcpack_workshop_2023)
  - Go to: day2\_orbital\_optimization
  - **Read README**
- 
- Tutorial problems:
    1. Converge energy w.r.t. basis
    2. Binding curve (likely on your own time!)



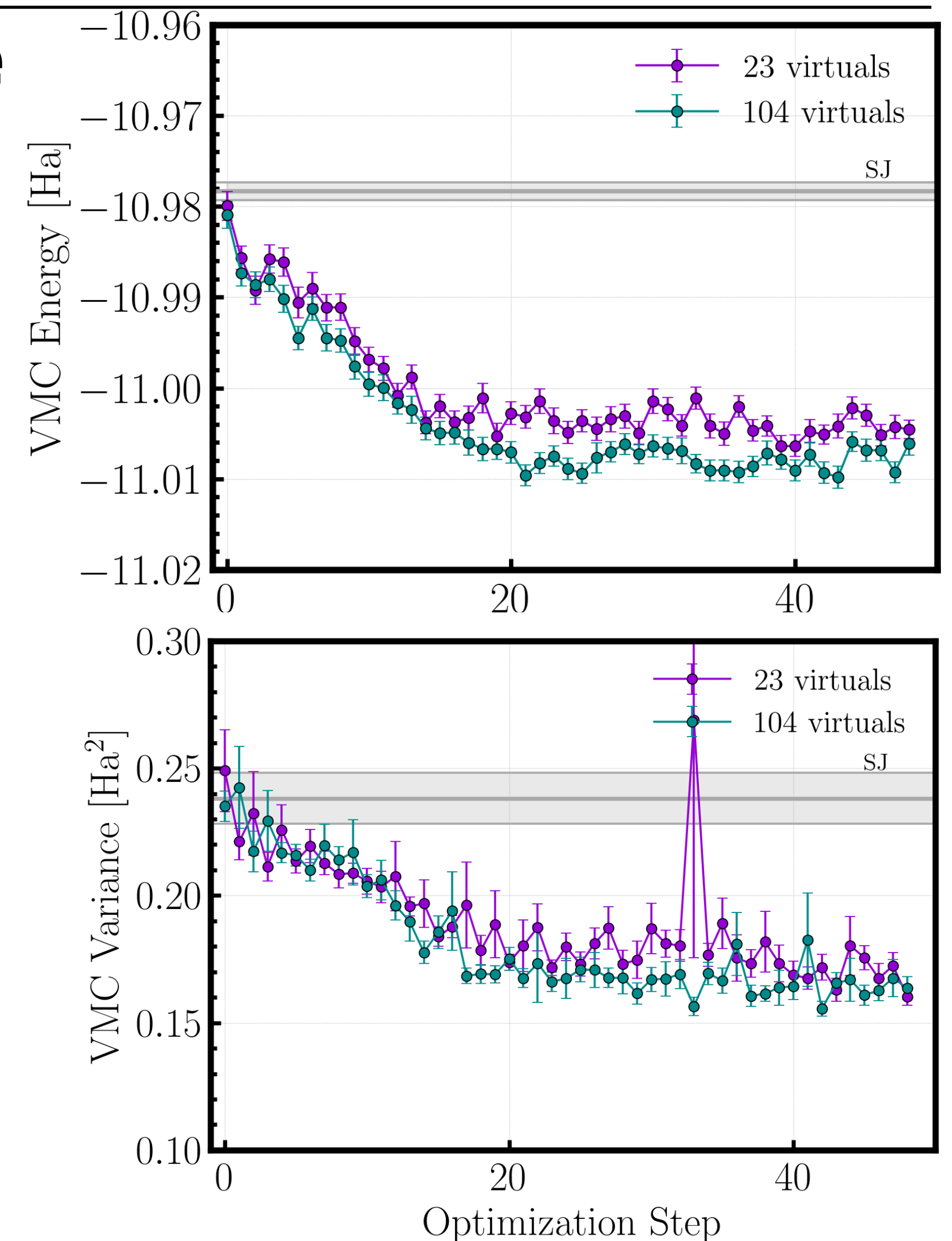
Carbon dimer geometry for problems 1 & 2.

# Tutorial: 0\_convergence

day2\_orbital\_optimization/0\_convergence

1. Go to '0\_orbital\_generation', run **psycf**
2. Go to '1\_qmc\_opt', run **qmcpack** in each `norbs_xyz` directory
3. Compare results (see example on right)
4. Go to '2\_qmc\_production', copy in optimized parameters, run **qmcpack**

You should find that more virtuals yields a lower energy - but only up to a point.

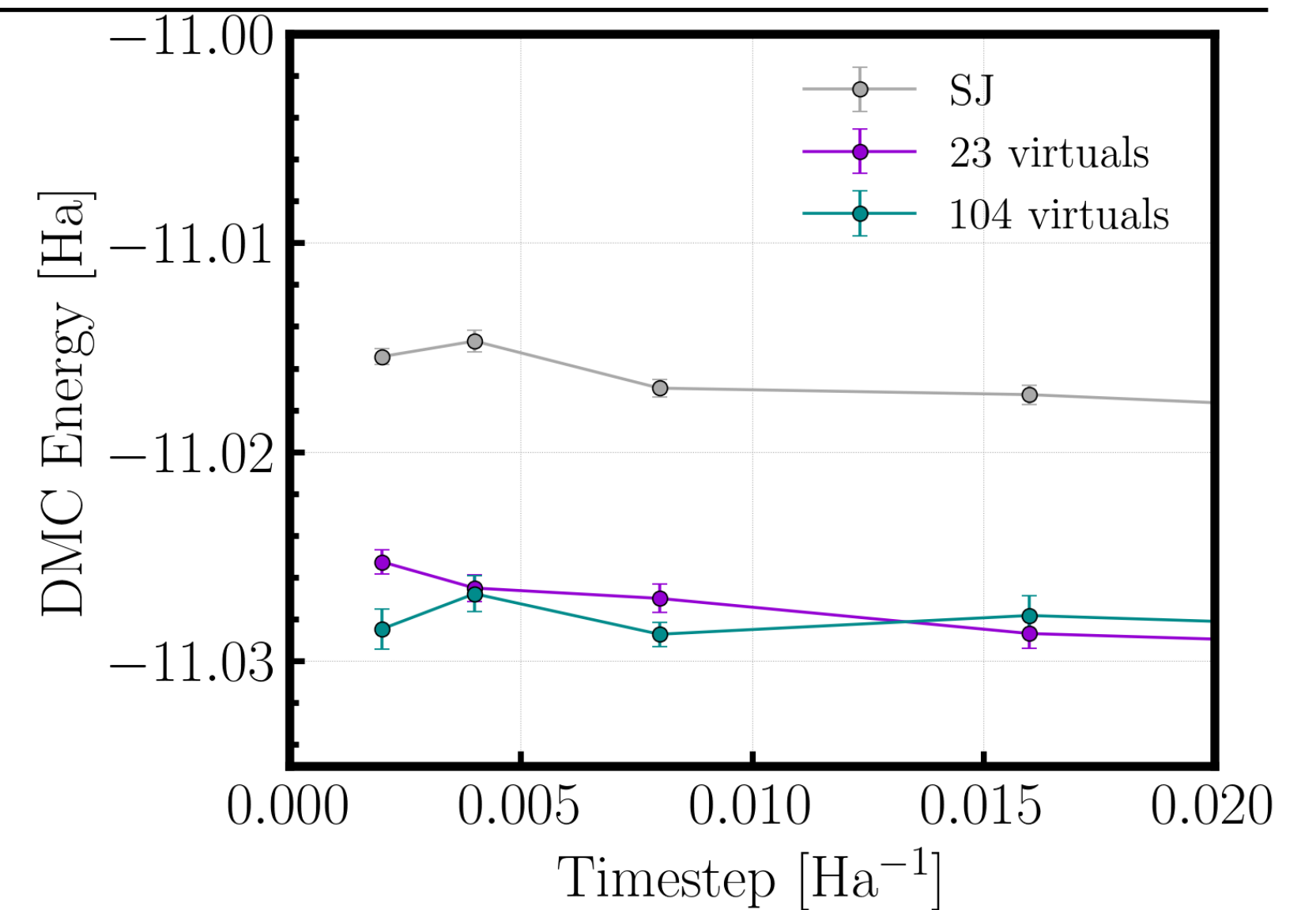


# Tutorial: 0\_convergence

## carbon-dimer/ecp/0\_convergence

1. Go to '0\_orbital\_generation', run **psycf**
2. Go to '1\_qmc\_opt', run **qmcpack** in each **norbs\_xyz** directory
3. Compare results (see example on right)
4. Go to '2\_qmc\_production', copy in optimized parameters, run **qmcpack**

You should find that more virtuals yields a lower energy - but only up to a point.



# Tutorial: 0\_convergence

---

carbon-dimer/ecp/0\_convergence

Suggested homework:

1. Adjust `minWalkers`, `shift_i` to see how optimization rate changes
2. Adjust `walkers_per_rank`, `steps`, `blocks` to speed up optimization w/ constant error bar
3. Try out binding curve calculation in `day2_orbital_optimization/1_binding_curve`

# References

---

1. Toulouse & Umrigar, JCP 126, (2007)
2. Toulouse & Umrigar, JCP 128, (2008)
3. Helgaker, “Molecular Electronic Structure Theory”, (2013)